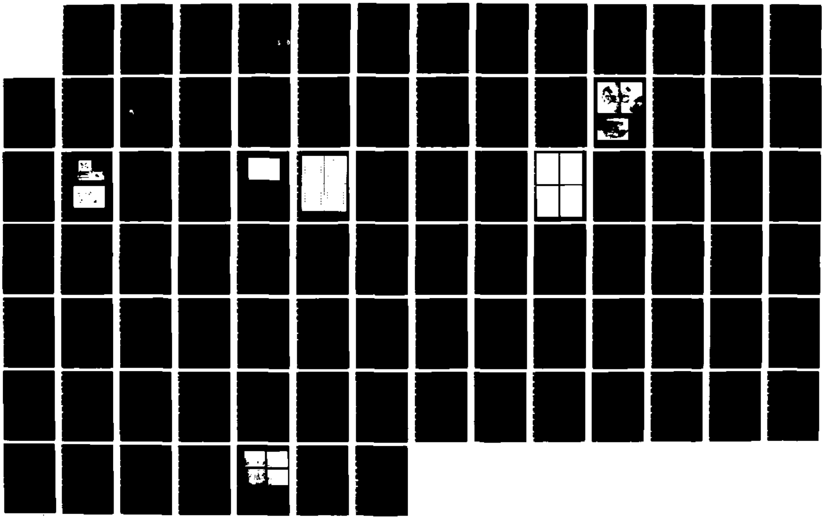
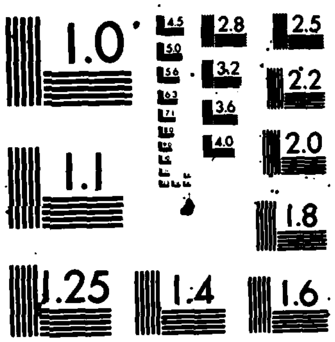
AD-A171 927 REQUIREMENTS AND APPROACHES FOR LOW COST SIMULATOR 1/1
GRAPHICS(U) SYSTEMS TECHNOLOGY INC HAWTHORNE CA
R W ALLEN ET AL JUN 85 STI-TR-1201-1 MDA903-83-C-0151

UNCLASSIFIED F/G 1/4 NL




1.0
4.5
5.0
5.6
6.3
7.1
8.0
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.8
1.1
1.25
1.4
1.6

2

REQUIREMENTS AND APPROACHES FOR LOW COST SIMULATOR GRAPHICS

1

SYSTEMS TECHNOLOGY, INC.

13766 SOUTH HAWTHORNE BOULEVARD • HAWTHORNE, CALIFORNIA 90250 • PHONE (213) 679-2281

Technical Report No. 1201-1

REQUIREMENTS AND APPROACHES FOR LOW COST SIMULATOR GRAPHICS

R. Wade Allen
Jeffrey R. Hogue
James C. Smith

June 1985

Contract No. MDA903-83-C-0151

Prepared for:

Director
Defense Advanced Research
Project Agency
1400 Wilson Blvd.
Arlington, VA 22209

[redacted]

BRANCH OFFICE: MOUNTAIN VIEW, CALIFORNIA

1. Report No.

2. Government Accession No.
AD-A171 927

3. Recipient's Catalog No.

4. Title and Subtitle
REQUIREMENTS AND APPROACHES FOR LOW COST
SIMULATOR GRAPHICS

5. Report Date
June 1985

6. Performing Organization Code

7. Author(s)
R. Wade Allen, Jeffrey R. Hogue, and James C. Smith

8. Performing Organization Report No.
STI TR-1201-1

9. Performing Organization Name and Address
Systems Technology, Inc.
13766 South Hawthorne Boulevard
Hawthorne, CA 90250

10. Work Unit No.

11. Contract or Grant No.
MDA903-83-C-0151

12. Sponsoring Agency Name and Address
Director
Defense Advanced Research Project Agency
1400 Wilson Blvd.
Arlington, VA 22209

13. Type of Report and Period Covered
Final Report

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

This report describes several advancements in low-cost simulation technology. The original intent of the project was to build a desk top F-16 Head-Up Display simulator. Although this ultimate objective was not achieved, advancements in computer generated display imagery and six degree-of-freedom vehicle dynamics were accomplished and are described herein. Commercial low-cost microcomputer and microprocessor components were used for all applications, and proved to have adequate computational capability for a variety of simulation tasks.

17. Key Words (Suggested by Author(s))
real time graphics
low cost graphics
computer generated imagery

18. Distribution Statement

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages

22. Price

For sale by the National Technical Information Service, Springfield, Virginia 22161

TABLE OF CONTENTS

Page

I. INTRODUCTION.. 1

A. Overview and Background.. 1

B. The Problem.. 1

C. Approach and Summary... 3

II. FUNCTIONAL DESIGN AND PHYSICAL LAYOUT............................. 4

A. Overview... 4

B. Functional Design.. 9

C. Prototype Architecture... 12

D. Physical Layout.. 12

III. DISPLAY DEVELOPMENTS.. 17

A. Overview... 17

B. Display Approaches... 17

C. Atari Graphics Computer and 8086 Interface....................... 22

D. Stores Control Panel (SCP)....................................... 22

E. Calligraphic Display Processor................................... 25

IV. HOST PROCESSOR CONFIGURATION AND COMPUTATIONS..................... 29

A. Overview... 29

B. Real-Time Dynamics... 29

C. Interface Software... 31

V. SUMMARY AND CONCLUDING REMARKS..................................... 34

REFERENCES.. 35

APPENDIX A. COMPUTATIONAL CONSIDERATIONS IN
REAL-TIME SIMULATION COMPUTER GRAPHICS.............. A-1

APPENDIX B. EFFECTS OF TRANSPORT DELAYS ON MANUAL
CONTROL SYSTEM PERFORMANCE.......................... B-1

APPENDIX C. A FAST, PROGRAMMABLE, LOW-COST DISPLAY
DEVICE FOR MAN-IN-THE-LOOP SIMULATION............... C-1

LIST OF FIGURES

Page

1. F-16 Cockpit Layout.......... 2

2a. Air-to-Surface Weapon Delivery Mode Selection/Displays.......... 5

2b. Dive Toss Task Description.......... 6

3a. Air-to-Air Weapon Delivery Mode Selection/Displays.......... 7

3b. Missile Override Task Description.......... 8

4. Functional Design.......... 10

5. Real Time Manual Control Loop.......... 11

6. Prototype Architecture.......... 13

7. Table Top Physical Layout.......... 14

8. Trainer Fifth Scale Mockups.......... 15

9. F-16 HUD Simulation Physical Layout Cockpit Configuration.......... 16

10. Atari Generated HUD Display.......... 20

11. Atari Graphics Computer HUD Display.......... 23

12. SCP Display Examples.......... 24

13. Calligraphic Display Processor System.......... 26

14. Calligraphic Display Scenes Including Horizon, Ground Grid, and HUD Pitch Scale.......... 28

15. F-16 Dynamics Computation Cycle Data Flow.......... 30

16. Run Time Benchmarks for Six Degree-of-Freedom Aircraft Dynamics.......... 32

A-1. Effects of Refresh Rate and Image Luminance on Critical Flicker Frequency (i.e., point at which image appears to have constant brightness). Ferry-Porter Law shown for photopic adaptation. Video, movie and CRT strobe written displays commonly set at 60 Hz refresh rate.......... A-5

LIST OF FIGURES (Concluded)

Page

A-2. Effects of Vehicle Angular Orientation on Perspective Display Plane Image Motion.................... A-7

A-3. Effects of Vehicle Translational Motion on Perspective Display Plane Image Motion.................... A-9

A-4. Simplified Crossover Model for Manual Vehicle Control and Stability Requirements (adapted from Ref. A-15).. A-11

A-5. Effect of Increased Effective Time Delay on System Bandwidth (Adapted from Ref. A-15)................. A-12

B-1. System Model for Air-to-Air Target Tracking............... B-3

B-2. System Model for Car/Driver Path Following a Commanded Path.. B-3

B-3. Generic Operator/Vehicle Tracking Dynamic Model for Analyzing Transport Delay Effects..................... B-4

B-4. Motion Feedback Closed-Loop Response Functions (Equivalent Closed-Loop Parameters Given in Table 1)....... B-8

B-5. Equivalent Motion Feedback Delay for Various Levels of τ_m and τ_c...................................... B-10

B-6. Equivalent Open-Loop Human Operator/Vehicle Describing Functions for Various Levels of Simulation Time Delays................................. B-12

B-7. System Bandwidth as a Function of System Time Delay.. B-15

B-8. Driver/Vehicle System Closed-Loop Response to an Impulse Disturbance.............................. B-17

C-1. Generic Model of a Human Operator/Vehicle Control Task to Illustrate the Effect of Computational Delays...... C-2

C-2. Computer/Processor Architecture for Calligraphic Display System.. C-5

C-3. Calligraphic Display Scenes Including Horizon, Ground Grid, and HUD Pitch Scale........................... C-7

LIST OF TABLES

Page

1. Low Cost Display Technology Assessment.................... 19

B-1. Motion Feedback Loop Parameters for Various Levels of Motion Feedback (τ_m) and Computational (τ_c) Delay................................... B-9

B-2. Human Operator/Vehicle Gain and Crossover Model Parameters for Various Combinations of Induced Vehicle/Simulator Delays....................... B-14

SECTION I

INTRODUCTION

A. OVERVIEW AND BACKGROUND

Systems Technology, Inc. (STI) contracted with DARPA to develop a low-cost device for training pilots in the use of the F-16 Head-Up Display (HUD). The HUD and associated subsystems are dominant features of the F-16's instrument panel, and are designed to be the pilot's primary aid in flight, weapon system energy management. In this role the HUD is an extremely versatile instrument, but by the same token is also quite complicated to use. Its three mode categories of flight management, ground attack and air combat include eleven different display formats with a bewildering repertoire of symbology. Training pilots to master the various HUD modes is the key to optimizing the F-16's deployment as an effective air superiority fighter.

B. THE PROBLEM

Air Force experience indicates that pilot trainees do not always receive enough flight time to learn the use of all the HUD display formats. A brief overview of the HUD system and its capabilities reveals the magnitude of the problem. An F-16 cockpit layout is shown in Fig. 1. HUD symbology is generally determined by selecting a mode via the stores control panel (SCP) shown in Fig. 1. HUD symbology can additionally be influenced by activating controls on the side stick and throttle handles, and the HUD control panel below the HUD display. Given the large number of display modes, and additional variations produced by optional control inputs, it is not surprising that a significant training problem exists.

Documentation also compounds the training problem. Procedures for the Navigation, Air-to-Air Combat and Air-to-Surface Attack modes are described in three documents, i.e., Refs. 1-3. These three documents must be carefully read and compared to gain a complete picture of the

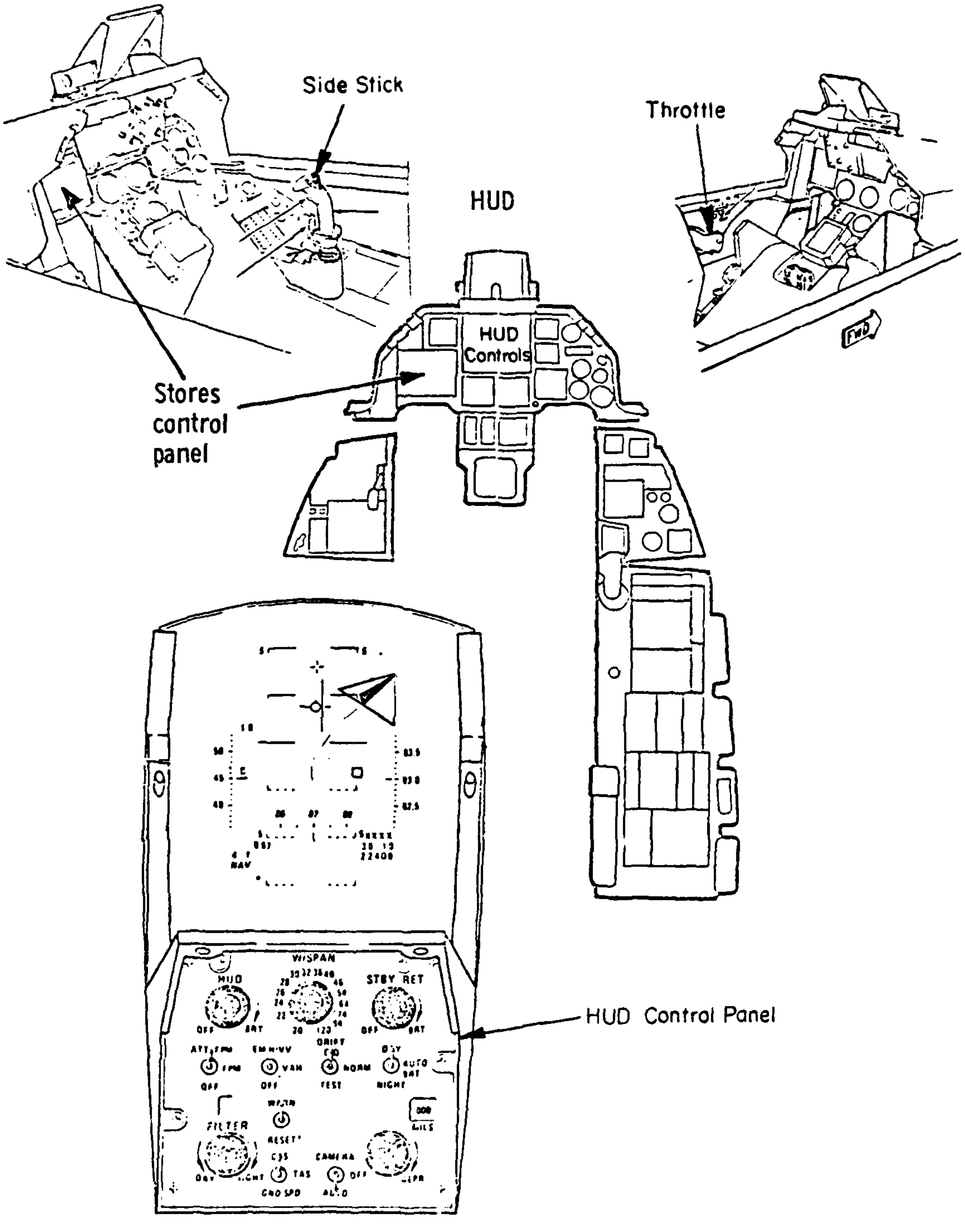


Figure 1. F-16 Cockpit Layout

HUD mode weapons delivery procedures, display options and energy management capabilities available to the F-16 pilot. Thus the overall training problem seems to result from a combination of complex procedures, a profusion of options and formats, and obscure and scattered documentation.

C. APPROACH AND SUMMARY

Based on successful experience with previous low-cost training devices, DARPA seeks to provide equipment that is highly motivating to operate and requires minimal support or training in its operation, much as is the case with commercial video games. The general approach taken here to accomplish the above objective was to configure a system with low-cost, commercial microcomputer and video hardware, and minimize any special-purpose hardware buildup. With this approach the functional requirements for the HUD tasks could be achieved primarily through software development. This approach also had the advantage of allowing easy modification or additions in the future to handle F-16 modifications and/or new training requirements.

Because of various development problems encountered during the project, it was not possible to complete hardware and software development with the available funds. Significant technical advancements were achieved in low-cost display technology, however, and valuable experience was gained that can be applied to future low-cost simulator developments. The nature of these developments will be discussed in the remainder of the report along with implications for using state-of-the-art technology in future low-cost simulator equipment. In Section II, the functional design and planned physical layout for the simulator are discussed. Visual display developments are discussed in Section III beginning with a review of current state-of-the-art low-cost technology, and including a discussion of functional requirements for display processing. Supporting details on display requirements are given in Appendices A and B. Supporting details on display advancements achieved in this project are given in Appendix C. Host processor considerations including dynamic computations are reviewed in Section IV. Finally, a technology summary and potential future advancements are summarized in Section V.

SECTION II

FUNCTIONAL DESIGN AND PHYSICAL LAYOUT

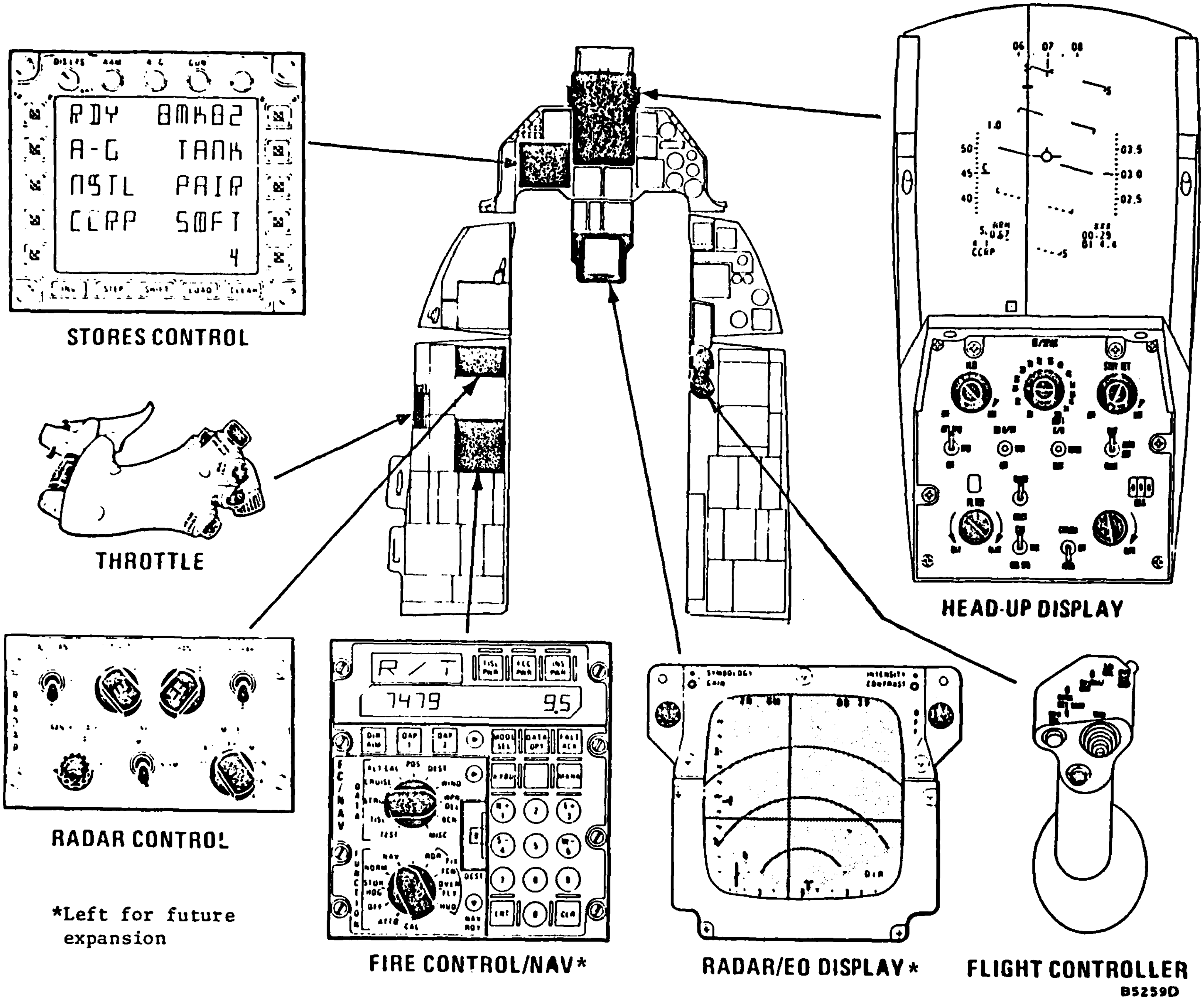
A. OVERVIEW

As part of the functional design effort, the pilot's usage of the F-16 HUD during typical flight maneuvers was reviewed in some detail in order to better define the required training hardware and procedures. As part of this review we also visited personnel at Wright Patterson AFB, Edwards AFB, Luke AFB, and Williams AFB to discuss procedures, and observe actual F-16 hardware and an F-16 simulation. It was generally concluded that the HUD-related tasks are complex and not clearly documented. The simulation group at Williams further pointed out that an enormous amount of work was required in setting up just the air-to-ground mode requirements for their simulation, in spite of the fact that they are working with an actual F-16 HUD.

The above state of affairs left us in much the same dilemma as F-16 pilots and the Williams AFB simulation group, i.e., a very complex system to learn with marginal resources to accomplish the task. This situation was recognized early on, so the following ground rules were set up to allow a versatile training device to be built without expending excessive work on superfluous details of system definition:

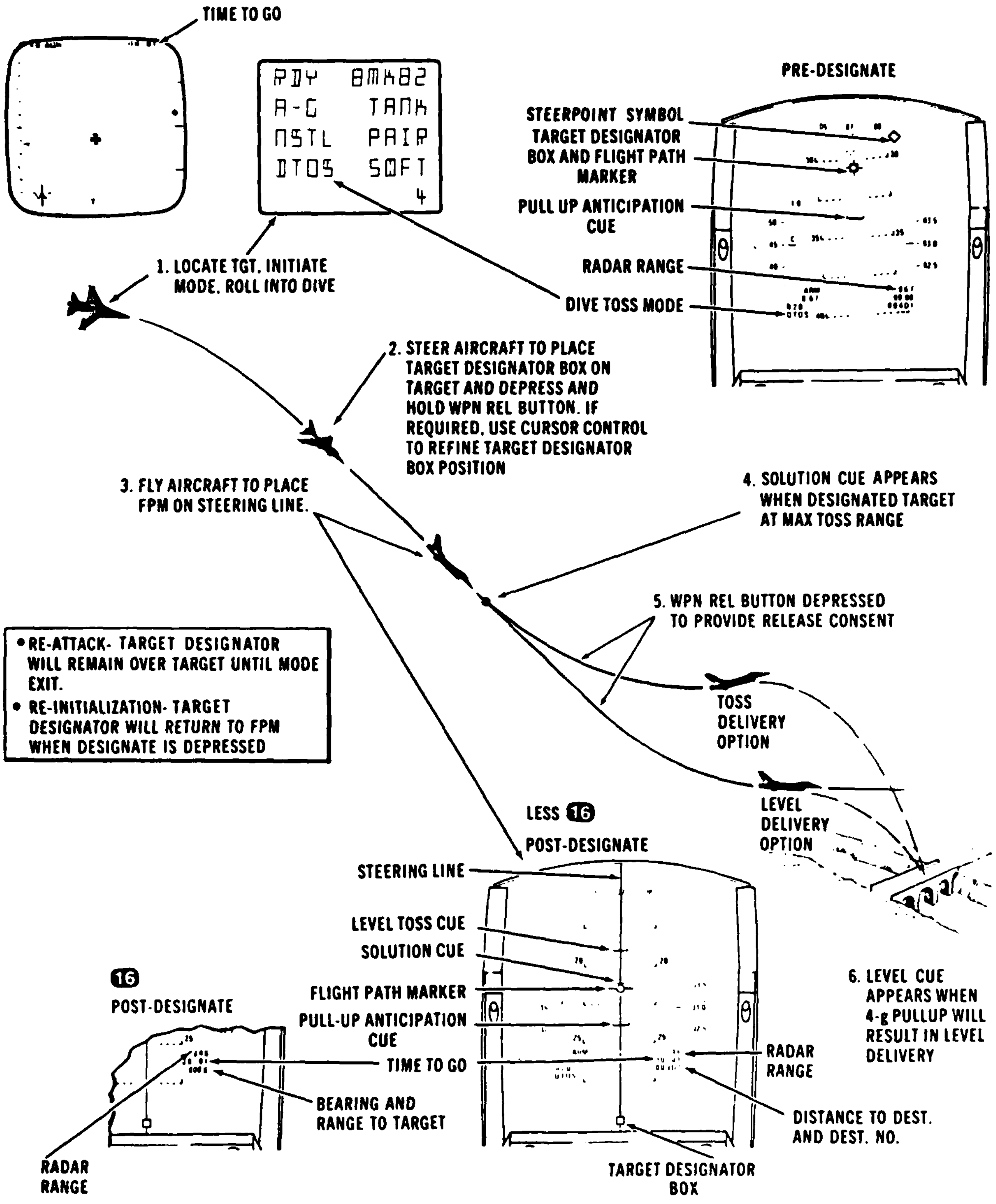
- The hardware configuration would be comprehensive enough to mechanize all HUD tasks.
- The computer software structure would also be comprehensive enough to handle all tasks.
- One air-to-ground and one air-to-air scenario would be selected as representative of the F-16's primary attack modes.

The air-to-ground and air-to-air tasks selected for mechanization are illustrated in Figs. 2 and 3, respectively. The hardware requirements for the air-to-surface and air-to-air weapon delivery modes are shown in Figs. 2a and 3a. Because of basic limitations imposed by a low



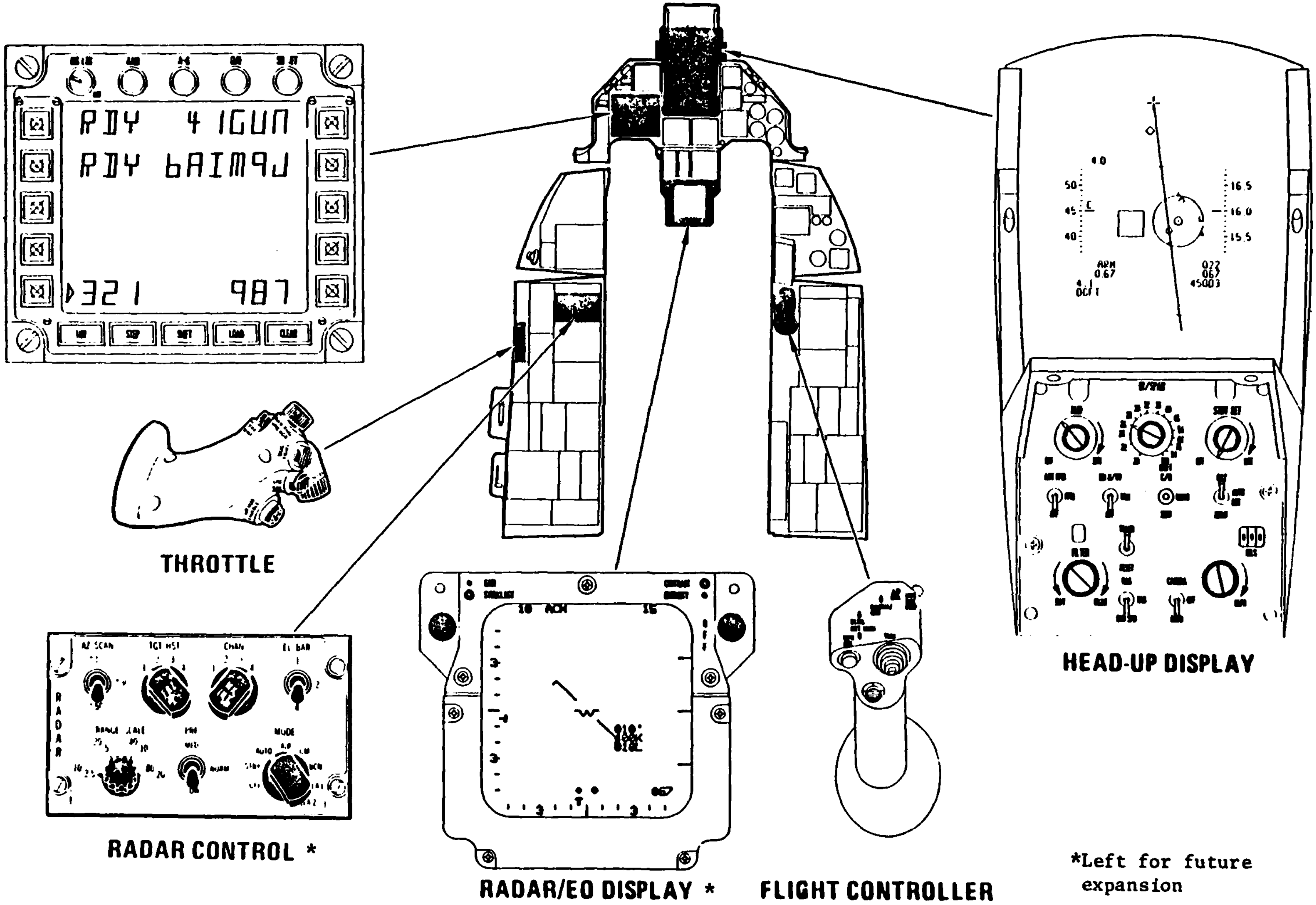
*Left for future expansion

Figure 2a. Air-to-Surface Weapon Delivery mode Selection/Displays


TIME TO GO
RDY 8MK82
A-G TANK
MSTL PAIR
DTOS SOFT
4
PRE-DESIGNATE
STEERPOINT SYMBOL
TARGET DESIGNATOR BOX AND FLIGHT PATH MARKER
PULL UP ANTICIPATION CUE
RADAR RANGE
DIVE TOSS MODE
1. LOCATE TGT. INITIATE MODE. ROLL INTO DIVE
2. STEER AIRCRAFT TO PLACE TARGET DESIGNATOR BOX ON TARGET AND DEPRESS AND HOLD WPN REL BUTTON. IF REQUIRED, USE CURSOR CONTROL TO REFINE TARGET DESIGNATOR BOX POSITION
3. FLY AIRCRAFT TO PLACE FPM ON STEERING LINE.
4. SOLUTION CUE APPEARS WHEN DESIGNATED TARGET AT MAX TOSS RANGE
5. WPN REL BUTTON DEPRESSED TO PROVIDE RELEASE CONSENT
• RE-ATTACK- TARGET DESIGNATOR WILL REMAIN OVER TARGET UNTIL MODE EXIT.
• RE-INITIALIZATION- TARGET DESIGNATOR WILL RETURN TO FPM WHEN DESIGNATE IS DEPRESSED
TOSS DELIVERY OPTION
LEVEL DELIVERY OPTION
LESS 16
POST-DESIGNATE
STEERING LINE
LEVEL TOSS CUE
SOLUTION CUE
FLIGHT PATH MARKER
PULL-UP ANTICIPATION CUE
TIME TO GO
RADAR RANGE
DISTANCE TO DEST. AND DEST. NO.
TARGET DESIGNATOR BOX
16
POST-DESIGNATE
BEARING AND RANGE TO TARGET
RADAR RANGE
6. LEVEL CUE APPEARS WHEN 4-g PULLUP WILL RESULT IN LEVEL DELIVERY

DTOS MODE

Figure 2b. Dive Toss Task Description



*Left for future expansion

Figure 3a. Air-to-Air Weapon Delivery Mode Selection/Displays

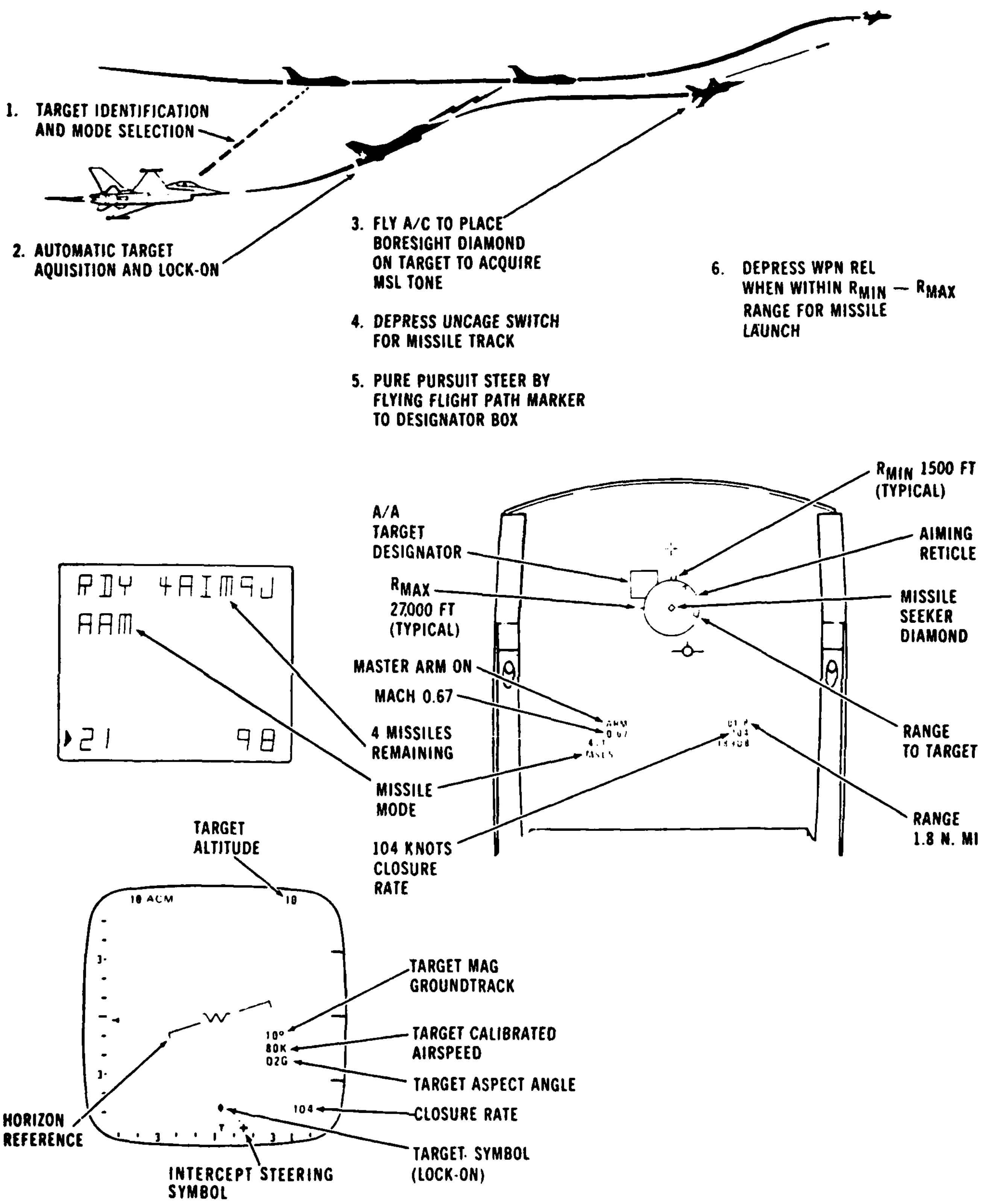

1. TARGET IDENTIFICATION AND MODE SELECTION
2. AUTOMATIC TARGET AQUISITION AND LOCK-ON
3. FLY A/C TO PLACE BORESIGHT DIAMOND ON TARGET TO ACQUIRE MSL TONE
4. DEPRESS UNCAGE SWITCH FOR MISSILE TRACK
5. PURE PURSUIT STEER BY FLYING FLIGHT PATH MARKER TO DESIGNATOR BOX
6. DEPRESS WPN REL WHEN WITHIN RMIN — RMAX RANGE FOR MISSILE LAUNCH
RDY 4AIM9J
AAM
▸21
98
A/A TARGET DESIGNATOR
RMAX 27,000 FT (TYPICAL)
MASTER ARM ON
MACH 0.67
4 MISSILES REMAINING
MISSILE MODE
104 KNOTS CLOSURE RATE
RMIN 1500 FT (TYPICAL)
AIMING RETICLE
MISSILE SEEKER DIAMOND
RANGE TO TARGET
RANGE 1.8 N. MI
TARGET ALTITUDE
10 ACM
10
10°
80K
02G
104
TARGET MAG GROUNDTRACK
TARGET CALIBRATED AIRSPEED
TARGET ASPECT ANGLE
CLOSURE RATE
TARGET SYMBOL (LOCK-ON)
INTERCEPT STEERING SYMBOL
HORIZON REFERENCE

Figure 3b. Missile Override Task Description

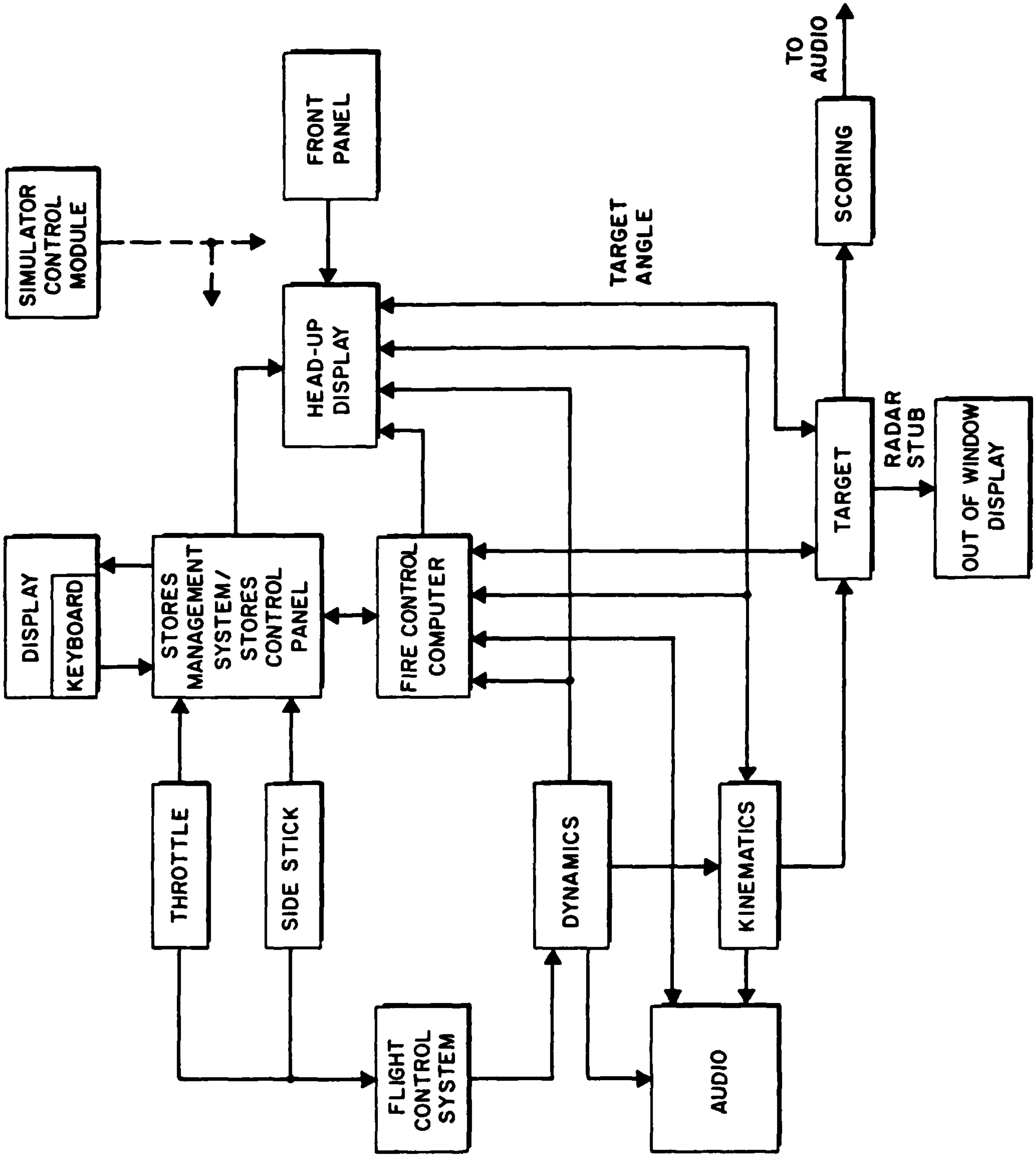
cost "desk top" physical configuration (discussed further on) we planned not to implement the starred components in Figs. 2a and 3a. These elements, according to EAFB and WAFB personnel, are somewhat peripheral to the primary modes in which the HUD is normally used. A key element in HUD operation is the stores control panel (SCP) which was added to the system that was originally proposed. A future option for the trainer could include a "sit-down" mockup (discussed further on) which could easily accommodate the starred items in Figs. 2a and 3a.

B. FUNCTIONAL DESIGN

An overall block diagram of major functional elements of a proposed F-16 HUD trainer is illustrated in Fig. 4. The various functions in Fig. 4 were planned to be mechanized with a combination of hardware and software. The objective of the functional design was to reflect DARPA and implicit Air Force requirements in achieving the stated objective: a low-cost, highly motivational F-16 HUD Training device.

Overall control of simulator operation in Fig. 4 is provided by the Simulator Control Module (SCM). This module exerts supervisory control over general simulator operations, based on simple menu-driven responses from the pilot. The SCM should provide structured instructions and mode options to the pilot allowing him to choose from appropriate HUD modes. Scenarios for the various HUD modes (i.e., navigation, ground attack, air combat) should be initiated and controlled by the SCM, via pilot commands received through the Stores Control Panel (SCP).

Scoring and feedback, the keys to motivating pilot participation, were also to be controlled by the SCM. Given various control task performance measures (e.g., tracking error, target hits, pullup or breakoff timing) the SCM could generate a composite score that rewards good performance and penalizes bad performance, which is then fed back to the pilot via visual and auditory displays. The details of the visuals can be structured so as to maximize the motivational impact of scoring. Similar approaches have been implemented in driving simulator research and have achieved a high degree of motiviation among test subjects (Ref. 5).


SIMULATOR CONTROL MODULE
FRONT PANEL
HEAD-UP DISPLAY
TARGET ANGLE
TO AUDIO
SCORING
DISPLAY
KEYBOARD
STORES MANAGEMENT SYSTEM/ STORES CONTROL PANEL
FIRE CONTROL COMPUTER
TARGET
RADAR STUB
OUT OF WINDOW DISPLAY
THROTTLE
SIDE STICK
DYNAMICS
KINEMATICS
FLIGHT CONTROL SYSTEM
AUDIO

The most critical portion of the functional design is the real time manual control loop shown in more detail in Fig. 5. In order to achieve realistic control dynamics, excessive computational delays must be avoided. Most critical is the control of aircraft attitude (pitch and roll on the HUD attitude bars and the out of window real world scene) and fire control symbols relative to targets. This "inner-loop" control must be updated at 20 Hz or faster to avoid unnatural control delay difficulties. The HUD symbology must also be presented with smooth apparent motion and without noticeable flicker. A refresh rate of at least 40 Hz will be required to achieve these objectives. Hardware and software selections for achieving these requirements are discussed below. Further detailed discussion of visual display requirements is given in Appendices A and B.

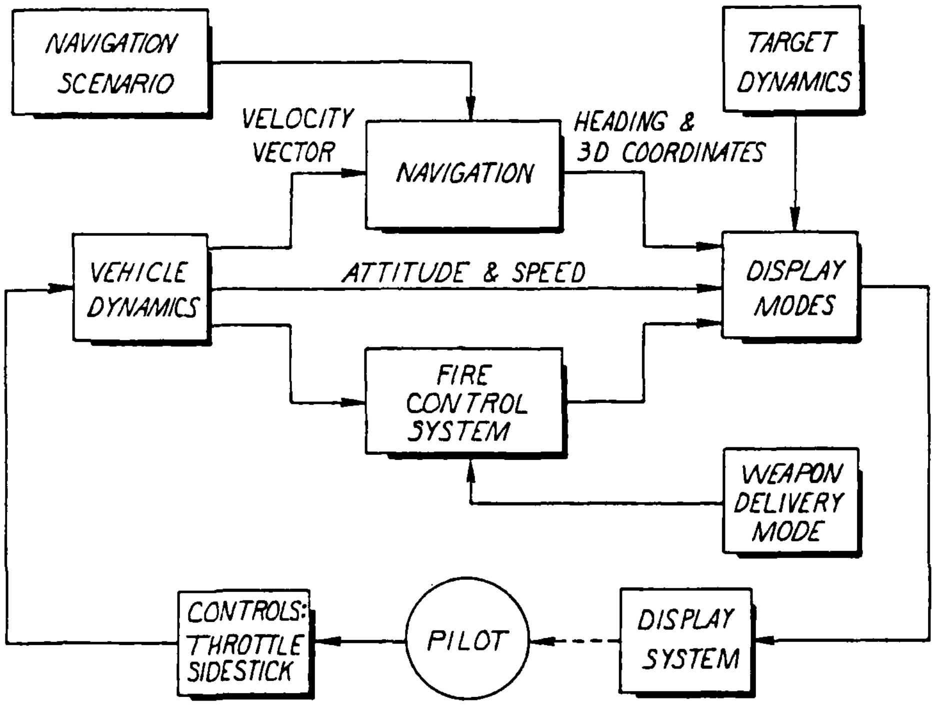


Figure 5. Real Time Manual Control Loop

C. PROTOTYPE ARCHITECTURE

Hardware decisions were based on computational requirements plus the need to simulate several F-16 displays. The HUD symbols and alphanumeric characters provided the most demanding requirement. A combination of home computer "video game" graphics plus a calligraphic display was chosen to meet these requirements, and the details are given in Section III. An Intel 8086 and multibus system were chosen to meet the real-time compution update requirement which must be accomplished at an update rate of 20 Hz or greater. The 8086 supports a variety of operating systems, e.g., CP/m-86, MS-DOS, and IAPX-86 thereby allowing the real-time dynamics to be implemented in a higher level language.

An Atari microcomputer was chosen to mechanize the HUD and SCP displays and auditory feedback. A "video game" computer such as the Atari is ideal for this task since it has special hardware with convenient software interface for creating video displays and sounds. An inexpensive color monitor or TV set can be used to provide the display and the audio system. A second processor combined with another inexpensive TV was selected for the stores control panel display. A block diagram showing a proposed prototype hardware configuration is given in Fig. 6. The use of the low-cost Atari system for the targeting display also established a framework for upgrading the color video system as new computer game technologies emerge.

D. PHYSICAL LAYOUT

Proposed physical layouts for a F-16 HUD trainer are illustrated in Fig. 7. The general configuration is designed to appear like the cowling and instrument panel of the F-16. The "outside world" display is mounted on top, while the HUD video monitor is mounted inside, and reflected off a mirror and the combining glass as illustrated. The remainder of the enclosure can be used for mounting the multibus card cage for the 8086 and HUD graphics boards, and also for mounting the Atari microcomputers. Photographs of mockups for both table top and freestanding physical configurations are shown in Fig. 8. Finally, a potential sit down configuration is shown in Fig. 9, which could include future enhancements such as radar displays.

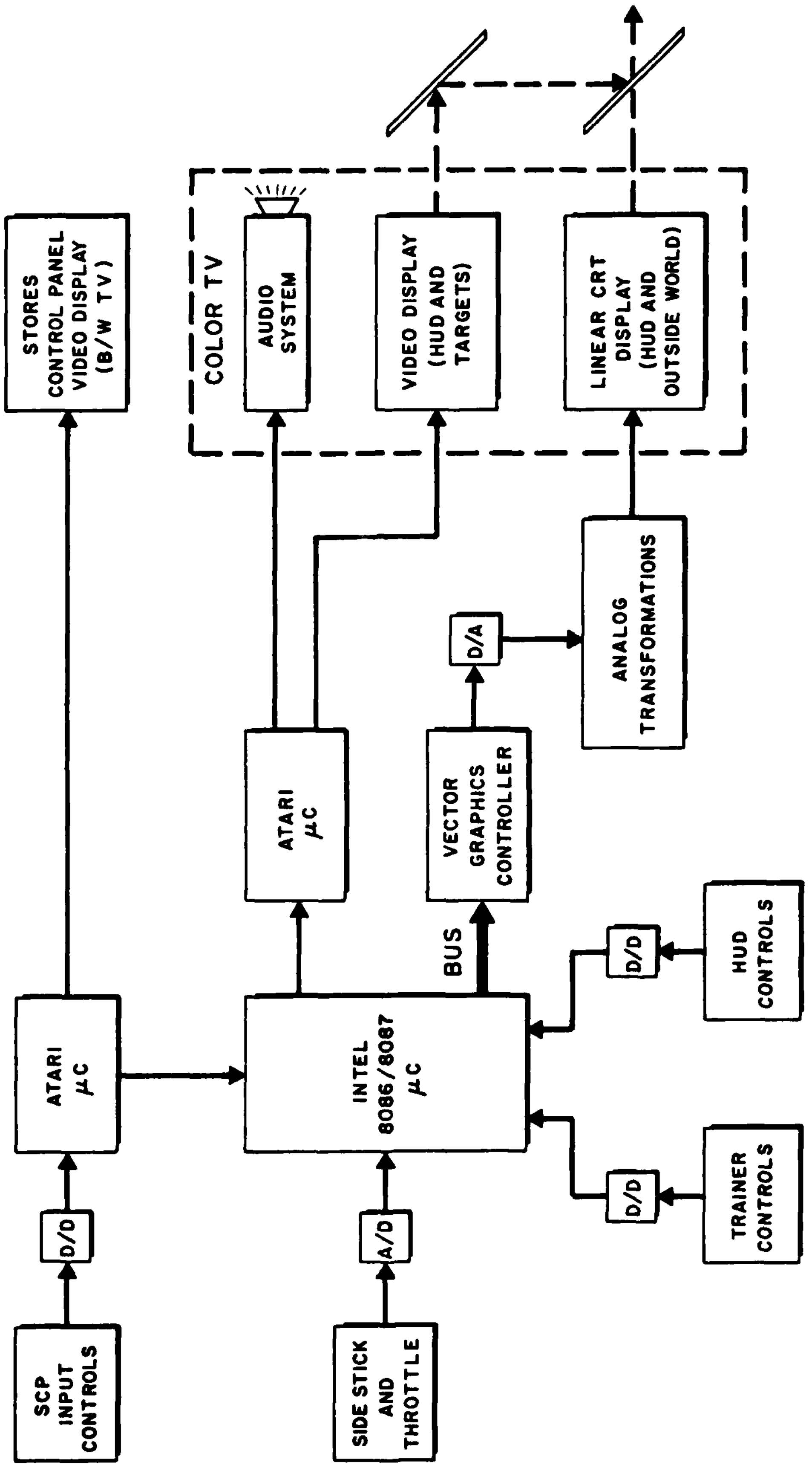


Figure 6. Prototype Architecture

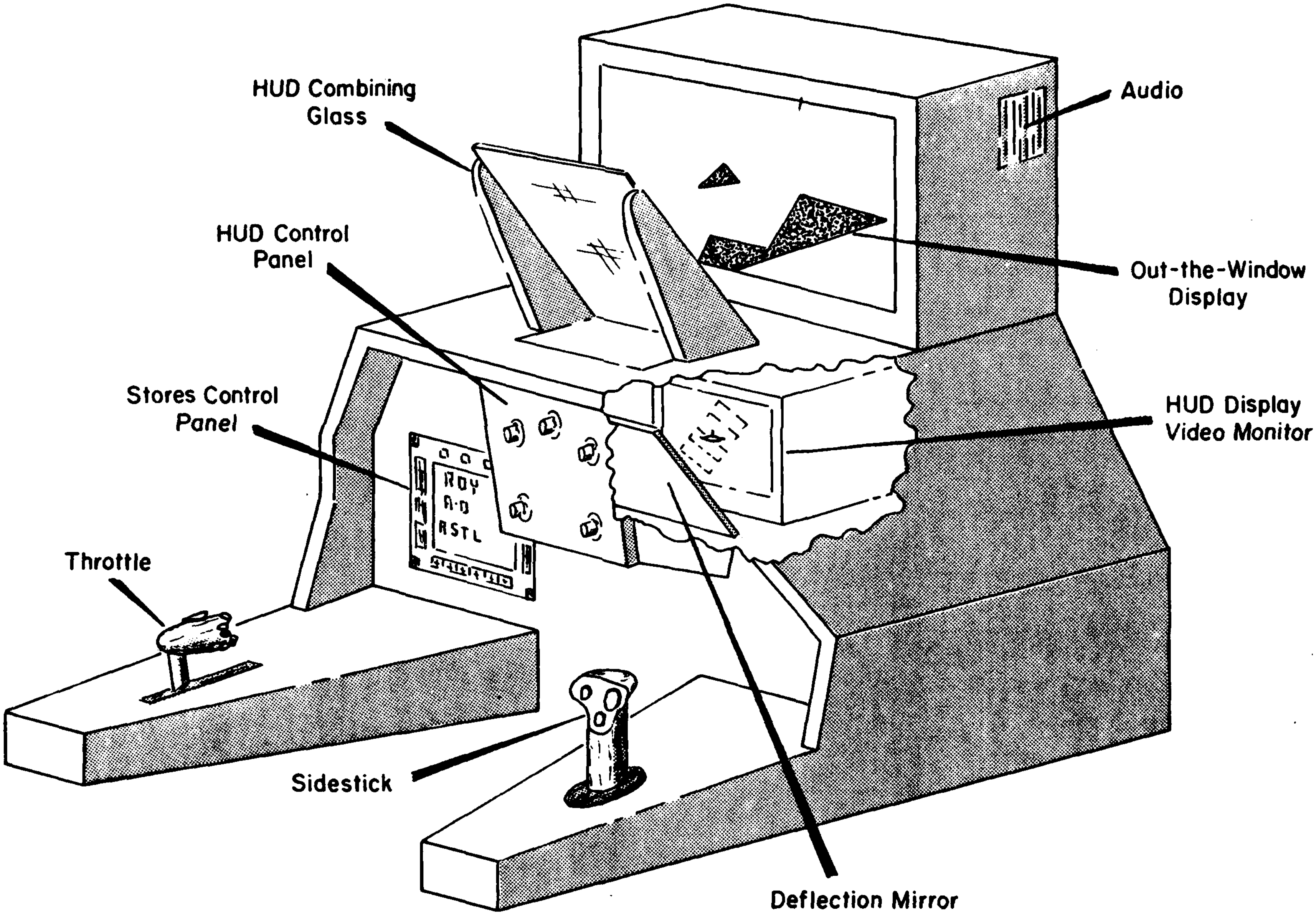
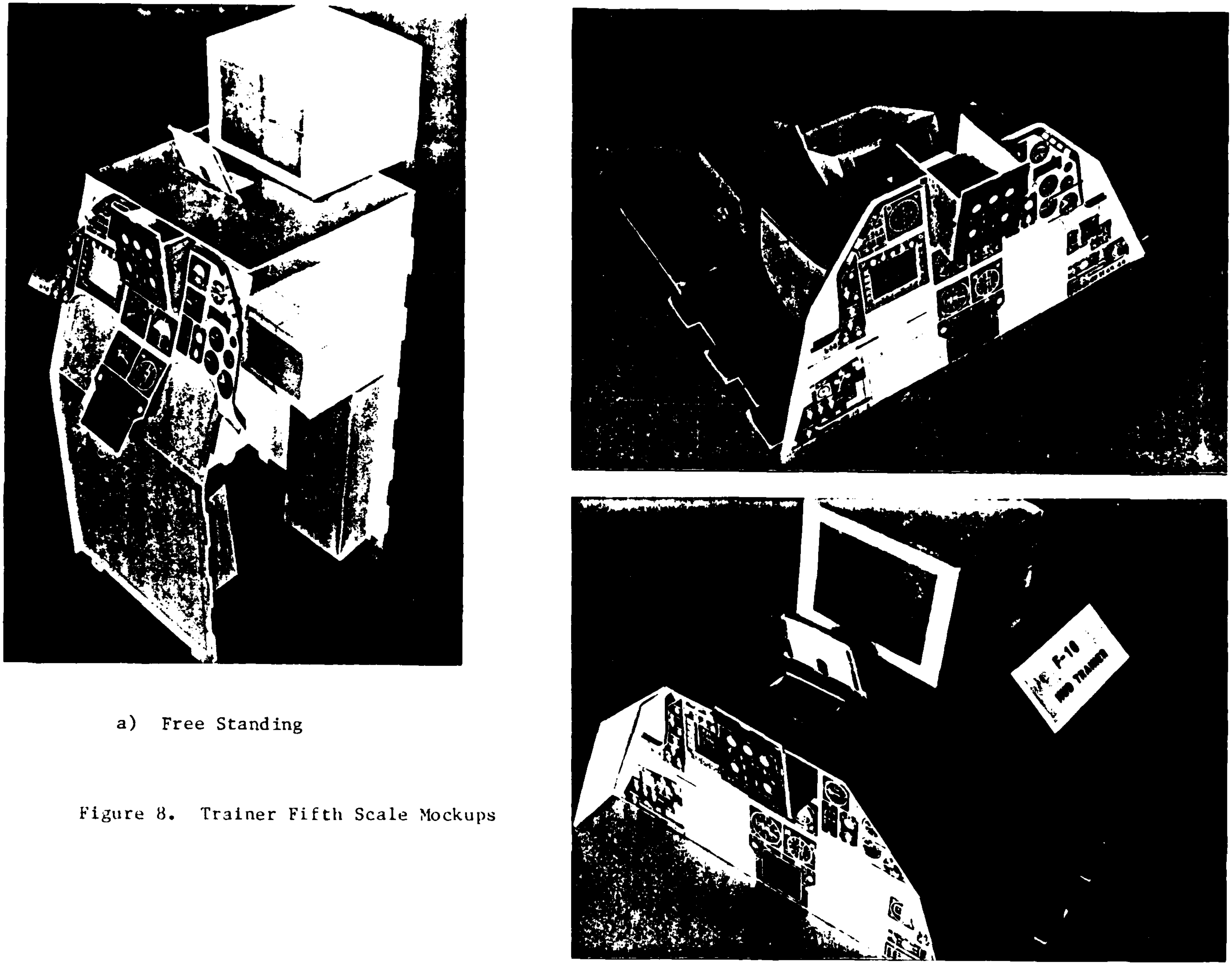


Figure 7. Table Top Physical Layout



a) Free Standing

b) Table Mount

Figure 8. Trainer Fifth Scale Mockups

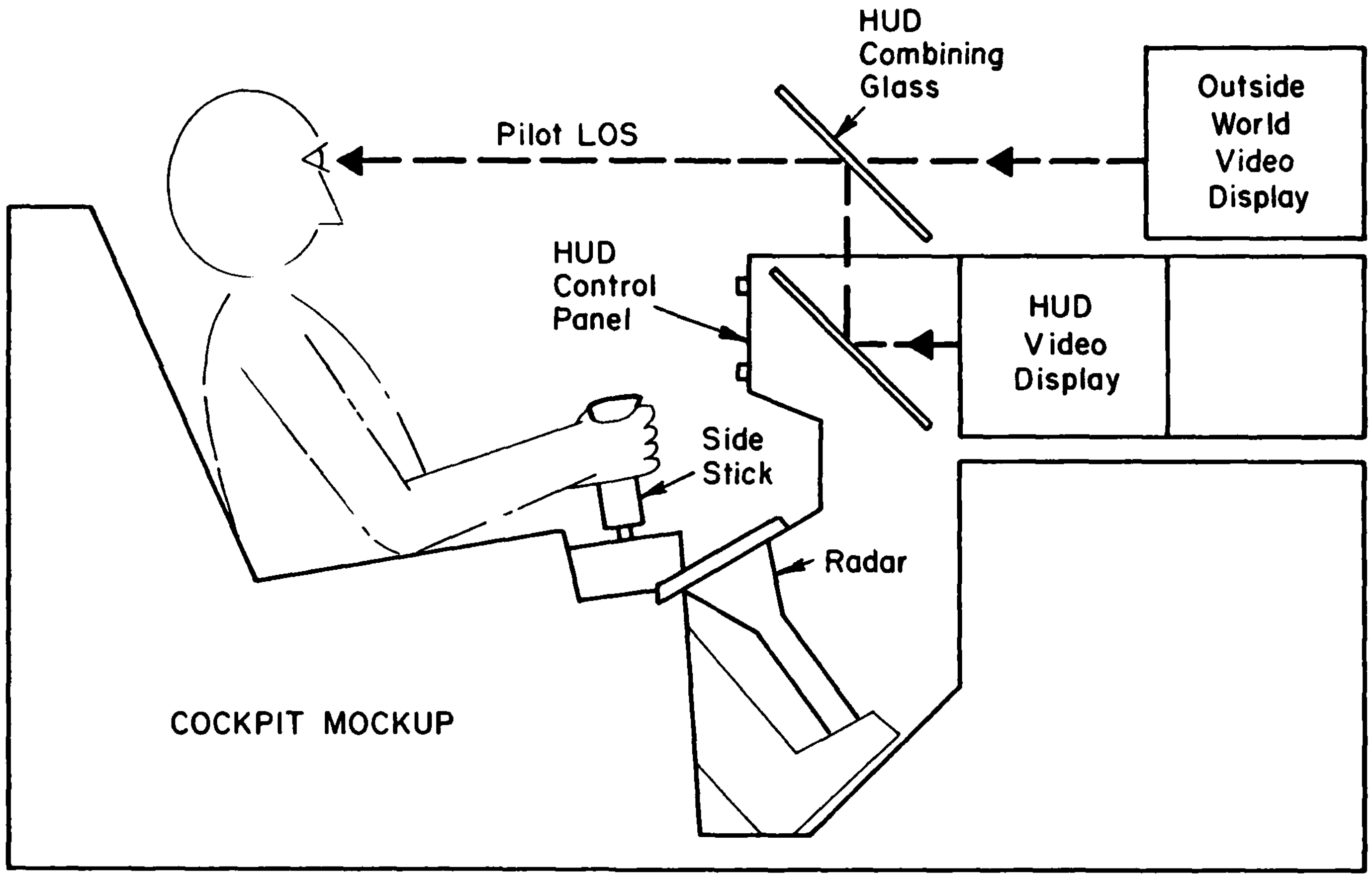


Figure 9. F-16 HUD Simulation Physical Layout Cockpit Configuration

SECTION III

DISPLAY DEVELOPMENTS

A. OVERVIEW

Graphics displays for meeting complexity and computational update requirements obviously presented the most difficult technical challenge on this project. Special graphics systems were obviously out of the question because of cost. Personal computer graphics systems appeared to hold the only prospect of providing a cost effective solution. Some home computers have developed special graphics IC's which permit overlaying objects on display fields through DMA like operations. Graphics boards developed for use with microcomputer systems also permit certain types of fast graphics processing. This effort started out with a survey of low-cost approaches. Two commercial systems were tried, and a special purpose system was developed on this project to meet the need for a fast update display generator that could portray six degree-of-freedom motion (i.e., roll, pitch, yaw; up, down, sideways). Details are given below.

B. DISPLAY APPROACHES

From a manual control and human factors point-of-view, dynamic flight displays must meet three requirements:

- they should appear to move smoothly
- they should not appear to flicker or flash which can be distracting and cause visual fatigue
- they should not present significant time delays which can cause the control tasks to seem unnaturally sluggish and potentially be uncontrollable

These requirements are discussed in some detail in Appendices A and B. Three display approaches were considered to meet the above requirements:

- Atari home computer system which uses a proprietary μ processor controller for animating a color raster display field and movable DMA (dynamic memory access) elements referred to as "sprites" or "players"

- A high resolution digital system for drawing vectors based on the NEC 7220 μ processor

- A digitally controlled analog system for drawing vectors on a linear CRT

A summary of the capabilities of these three systems is given in Table 1. Initially we attempted to mechanize both the HUD display and out-the-window real world scenes in an Atari home computer. The Atari has proven excellent for allowing rapid movement of display elements in a rectangular orientation as will be discussed subsequently. Figure 10 shows an Air-to-Air mode display which allows movement of several reticles and linear scales. The interface has been worked out to drive the HUD display elements from the system's 8086 host processor.

The Atari was also used to mechanize the SCP (Stores Control Panel) display as discussed in III-D. The HUD and SCP mechanizations have two basic characteristics in common that make them ideal for the Atari application: they require minimal computation and they basically have a rectangular orientation.

In our attempts to develop an out-the-window display, we were able to simulate perspective scene motions relative to pitch, yaw, and lateral translation motions of the observer. Because of the slow computational speed of the Atari host processor (a Mostek 6502 running on a 2.8M Hz clock) rolling the display in real time would be virtually impossible, and adding other degrees of freedom would be prohibitive in terms of assembly language development. The out-the-window scene developments indicated that the Atari would be useful for ground vehicle displays which do not have to roll or change altitude (relative eye height). Some arcade and home video games have in fact demonstrated this capability (e.g., "Pole Position").

TABLE 1. LOW COST DISPLAY TECHNOLOGY ASSESSMENT

Atari

- Good for rectangular oriented formats
- Can animate players and play field
- Rapid, smooth slewing
- Cannot do roll
- Must minimize computations
- Color priorities, hit detection
- Sound

[Good, fast, low-cost rectangular graphics system]

NEC 7220-μP Systems (Ikier, Vectrix, NEC, ...)

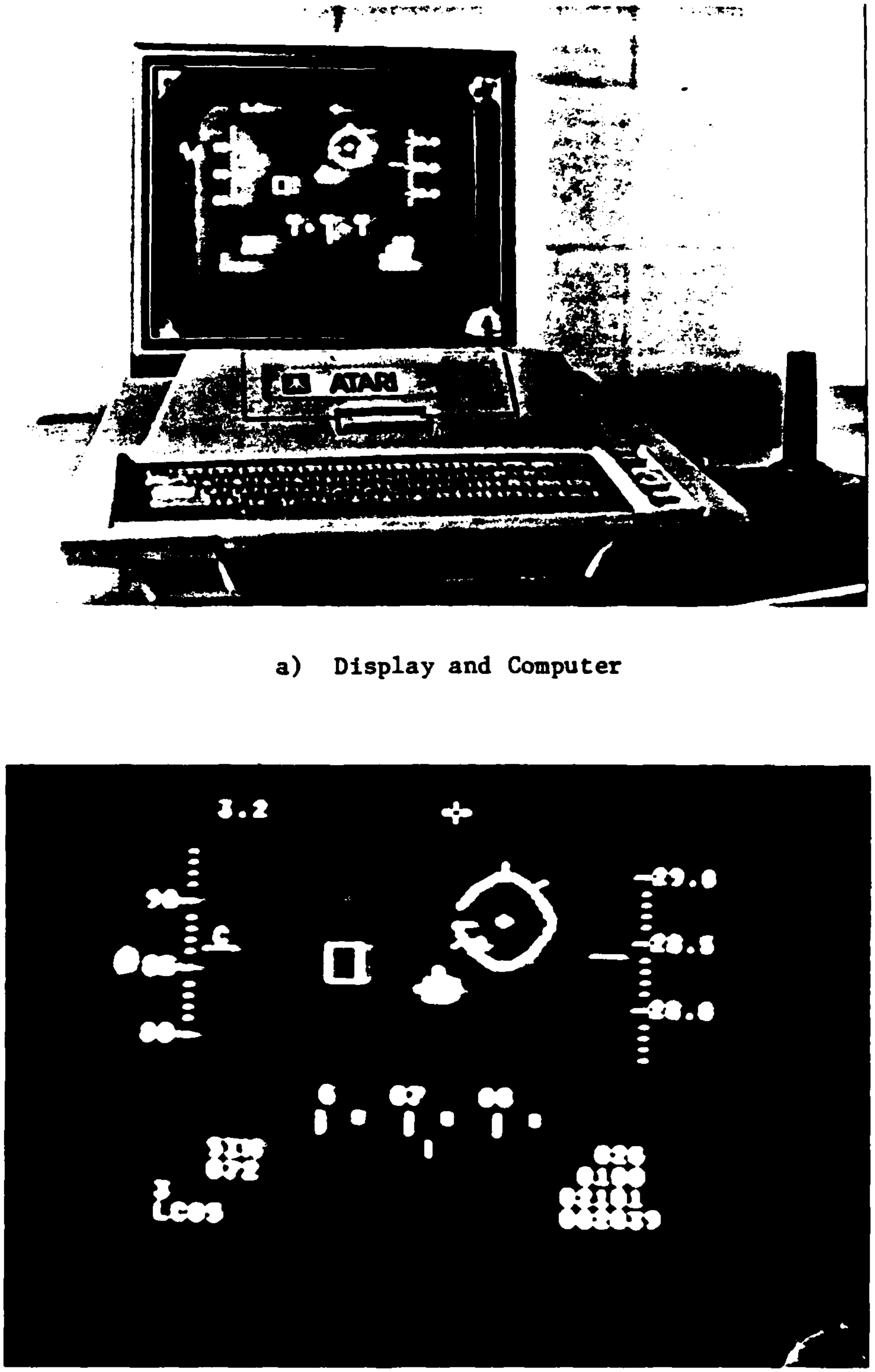
- Significant overhead to load chip with instructions
- Typical vectors require ~ 5 msec for draw and erase
- Circles require 8 vectors
- For 20 Hz update rate can generate 10 vectors

[Not adequate for complex real time dynamic displays]

Calligraphic systems

- > 100 vectors in non flash mode
- can perform analog transformations with no delay penalty
- High resolution, smooth movement

[Best approach for moderate cost, complex scene generation]



a) Display and Computer

b) Display Close-up

Figure 10. Atari Generated HUD Display

The second system we investigated employed a NEC 7220 μ processor to generate vectors with 800 × 1024 pixels on a digitally (TTL) driven display. The NEC chip has several built in modes which allow for drawing arcs, vectors, and filling in polygons. This is a very appealing approach and a number of digital graphics and μ computer systems are now using the NEC chip as a graphics controller. Our own experience has shown the NEC chip to be too slow for real time display applications, however. Two NEC based systems were benchmarked. One included in an 8086 development system acquired for this project, and another in incorporated in a different commercial product.

An array of 15 bytes must be loaded into the NEC chip for every vector draw, 17 for each arc sector and rectangle and 7 for a single dot. For animation, vectors must be erased and redrawn at a new location. A further limitation is that the maximum arc sector only extends for 45 deg, so a full circle requires 8 consecutive arc sector loads or 136 byte transfers plus housekeeping. We found that about 5 msec were required for a draw and erase, so that even at a fairly slow update rate of 20 Hz only about 10 vectors can be generated (barely enough for one circle!). It should be noted that the specifications for the NEC based systems do not relate to this update issue. Their applications are intended more towards CAD/CAM where several seconds might be adequate for a display update.

The above low cost approaches are clearly not adequate for real world out-the-window scenes, and for this we decided to return to the third approach shown in Table 1, a calligraphic or analog waveform system drawn on a linear CRT. The attitude and translation transformations required for real world perspective motion can be accomplished with analog multipliers which essentially work with no computational delay. This approach has been used quite successfully in the past for ground vehicle simulator displays (Ref. 5). To make this approach more general, however, the waveforms for the basic three dimensional map were generated digitally. The digital waveform generation technique has been developed previously (Ref. 6) and we were able to test this approach with our transformation system. Our feasibility tests showed that over

100 arbitrary vectors could be generated, transformed, and displayed with high resolution, smooth motion, no flicker at a fairly high update rate (~ 50 Hz).

Based on the above assessments, we decided to use the Atari graphics approach for HUD rectangular graphics and the SCP display. The calligraphic approach was used for real world out-the-window scenes and the HUD attitude bars which must rotate with the aircraft's roll angle as well as translate for pitch.

C. ATARI GRAPHICS COMPUTER AND 8086 INTERFACE

A closeup of the air-to-air mode display implemented on the Atari is shown in Fig. 11. In this display, there are three symbol categories that must be updated with information from the 8086 host processor:

- linear moving scales: vertical moving airspeed and altitude scales, and the horizontal heading scale
- targeting symbols with vehicle and horizontal movement: air-to-air aiming reticle, target designator box, and flight path marker
- alphanumeric information in 15 "windows:" e.g., vertical g's, heading, weapons mode, etc.

Two parallel ports are used to pass alphanumeric and movement information for the above symbols between the 8086 and the Atari. The information is contained in a 30 byte block. Under free running conditions we have achieved a data rate of 6000 bytes/sec. In order to be able to pass all required information during the Atari's vertical blanking interval, the display update rate is run at 30 times a second. The Atari still maintains a 60 Hz video refresh rate, however, so that no flicker is evident. Different portions of the display are updated on alternate frames.

D. STORES CONTROL PANEL (SCP)

As discussed previously, the SCP was not included in our original concept for the F-16 HUD trainer, but subsequent discussions with pilots

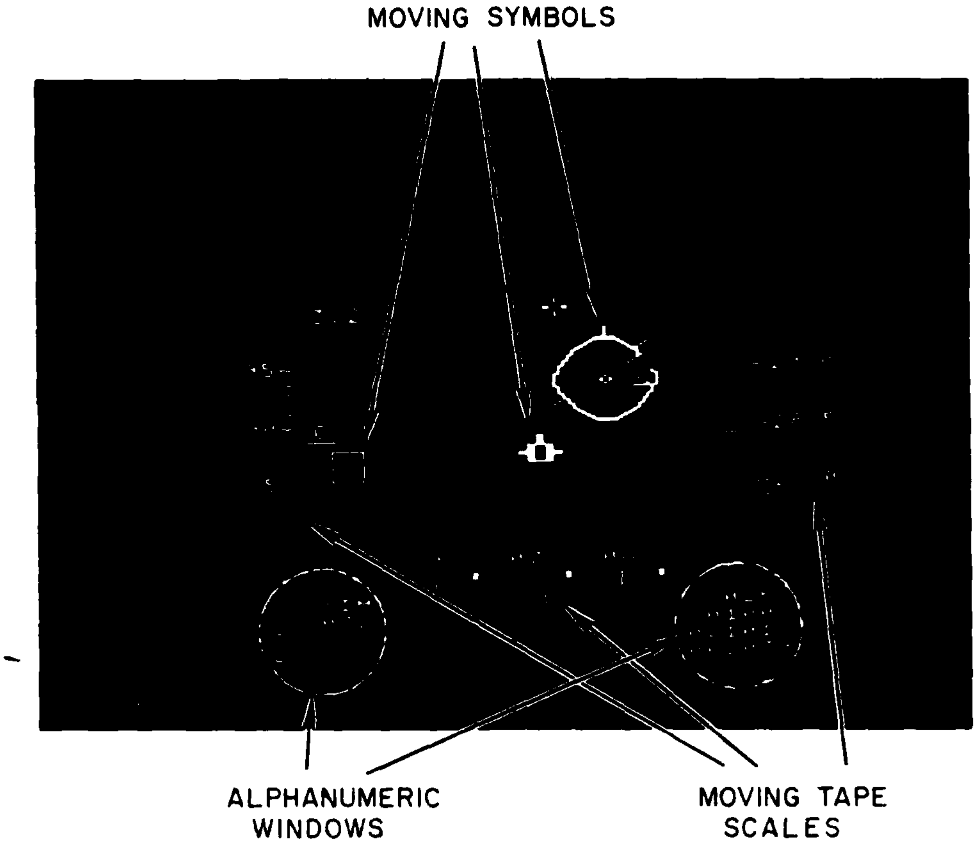


Figure 11. Atari Graphics Computer HUD Display

and training simulation personnel implicated it as an integral part of the HUD training problem. Preliminary analysis of the SCP's functional requirements proved it to be well suited for simulation on an Atari microcomputer. Modification of the standard Atari character font allowed us to simulate the special 9 segment panel characters on an ordinary TV set.

The logic for the SCP display and keyboard input were implemented in Atari Basic. Several typical SCP display formats are illustrated in Fig. 12a. The displayed buttons were subsequently replaced by a switch panel as shown in Fig. 12b which interfaced with the Atari keyboard input port. Most of the weapon modes were represented in the software, which could easily be updated or augmented to accommodate additional or altered weaponry.

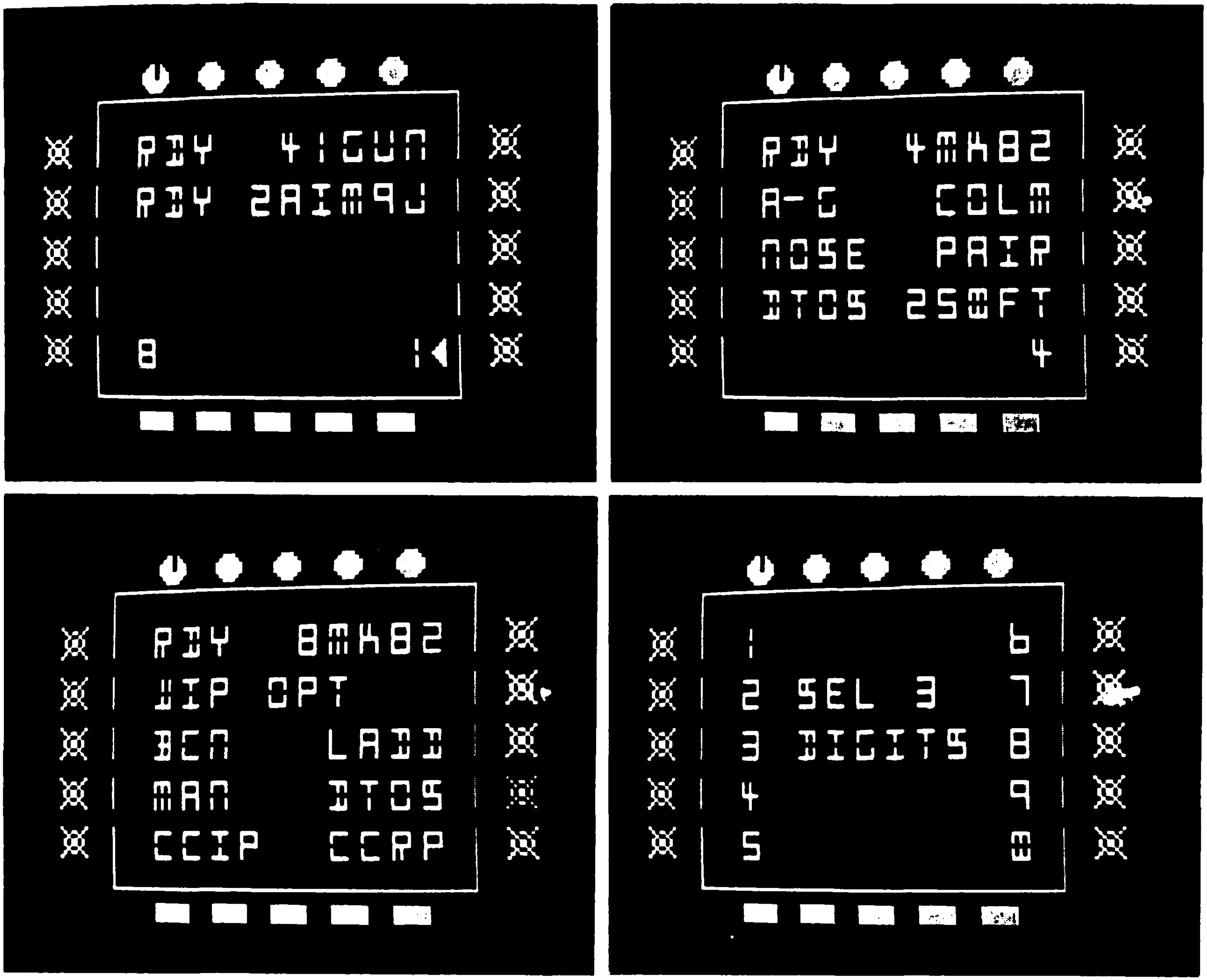


Figure 12. SCP Display Examples

In order to have the SCP automatically boot on power up, we intended to compile the Atari Basic and then store the object code on a ROM. We were successful in putting the HUD graphics code on a ROM, but the SCP effort was not completed.

E. CALLIGRAPHIC DISPLAY PROCESSOR

The calligraphic processor was designed as a low cost solution to the problem of rapid manipulation of arbitrary vectors on a linear x-y CRT. The processor consists of two major elements: 1) a digital waveform generator which draws three-dimensional (rectangular coordinate) maps relative to observer position from memory stored coordinates; 2) an analog transformation system which provides angular transformations for orienting the observer's line of sight and a perspective transformation for drawing images within the observer's field of view on the display plane.

A block diagram of the overall processor system is shown in Fig. 13. The digital waveform generator is a single multibus card which plugs directly into the 8086 bus. The card contains a Z80 microprocessor, 64K bytes of memory (an arbitrary mix of RAM and ROM), and hybrid circuitry which generates x, y, and z axis analog waveforms, associated blanking pulses and an intensity control waveform. The 8086 can write directly into the Z80 memory space to update the observer's viewing position (x, y, z coordinates) within the three-dimensional map. Memory conflicts are arbitrated by a hardware bus lockout when the Z80 is reading, and by a shared software semaphore flag which the 8086 uses when it writes address data into Z80 RAM. The computational load on the Z80 is minimal, consisting of memory fetch and a fixed point add to update the observer position for each vector and point. Thus the vector rate of the processor is relatively high, running about 8000 vectors per second. At a 50 Hz update rate to minimize flicker, we can nominally present display scenes with about 160 vectors. (For future upgrades, a faster 16 bit processor would allow more vectors and finer resolution.)

The digital waveform generator sends x, y, z, and intensity signals in the form of a three-dimensional vector to an analog transformation

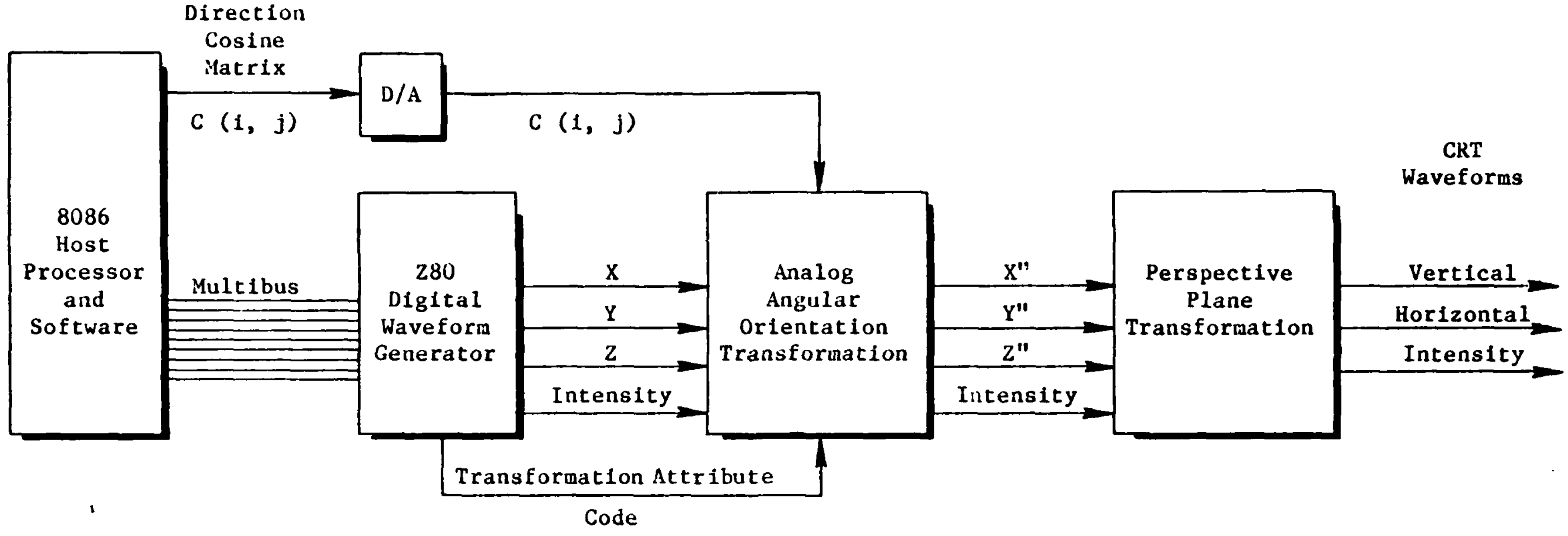
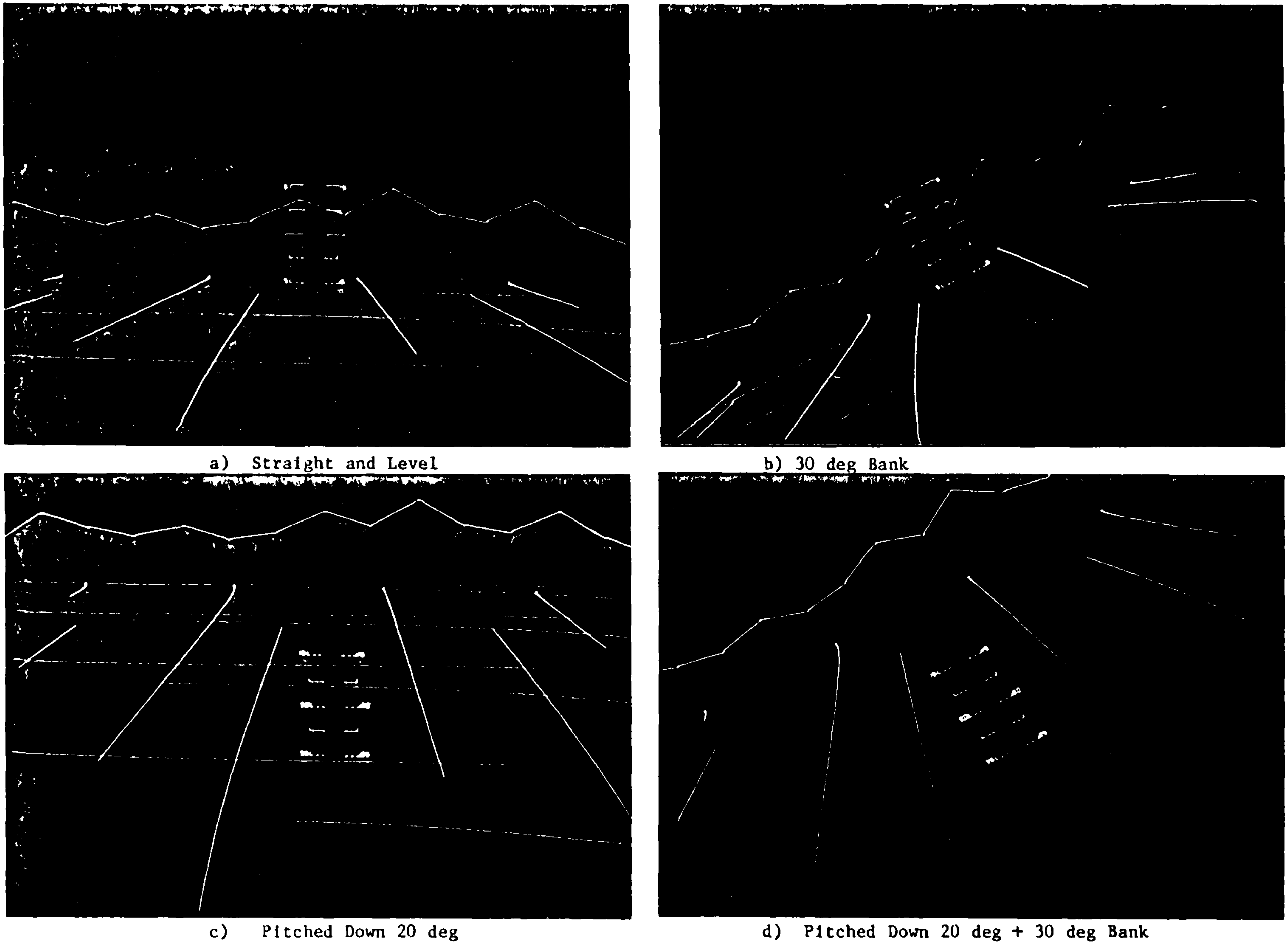


Figure 13. Calligraphic Display Processor System

processor (see Fig. 13), which consists of angular and perspective transformation circuitry. The angular transformations are driven by the vehicle direction cosine matrix derived from equations of motion in the 8086 host. They are then converted to analog voltages and transmitted to the transformation processor via auxiliary cable. This approach guarantees that angular motions exhibit minimal display processor lag. Minimization of angular response lags is crucial in flight simulation. Since a pilot's broadest bandwidth requirements are in attitude control, which is highly sensitive to computational delay.

One powerful feature of our approach to display generation is that each vector is individually tagged for the categories of transformations to be applied to it by the transformation hardware. As noted in the Fig. 13 block diagram, the Z80 microprocessor sends a transformation attribute code with each vector. The attribute code is then used by the transformation processor to select which of several sets of transformations will be applied to the vectors. For example, this approach allows us to generate HUD pitch scales which only require roll transformation, airplane referenced vectors (e.g., gun tracers) which only go through the perspective transformation, and perhaps other HUD symbology that does not require any transformation at all.

Photos of several display configurations are shown in Fig. 14. This graphics approach allows for relatively complex programmable scene generation with low computational delays. The hardware is inherently low cost, and allows for additional future enhancement of speed and capacity. This type of calligraphic system could potentially provide for curved vectors (not just straight line approximations) and limited fill capability. Also, some color capability could be added by using a shadow mask, stroke-writing linear CRT. Further discussion is given in Appendix C.



a) Straight and Level

b) 30 deg Bank

c) Pitched Down 20 deg

d) Pitched Down 20 deg + 30 deg Bank

Figure 14. Calligraphic Display Scenes Including Horizon, Ground Grid, and HUD Pitch Scale

SECTION IV

HOST PROCESSOR CONFIGURATION AND COMPUTATIONS

A. OVERVIEW

As noted in the prototype architecture of Fig. 6, an Intel 8086 microprocessor with an 8087 math co-processor was selected for the central host processing chores. FORTRAN software and some assembly language subroutines were run under the CP/M 86 operating system. As discussed in Section II-C the host processor was designated to handle the overall housekeeping functions and carry out the computationally intensive real-time computations. In this role the host processor had to provide drive commands for the Atari HUD and the out-the-window vector graphics system, and accept commands from the Stores Management Subsystem, stick, and throttle. Details on successfully completed parts of the host system software were as follows.

B. REAL-TIME DYNAMICS

The Intel 8086 system was configured to run under CP/M-86 which, in turn, was configured to support a subset of iAPX 86,88 which was originally developed for the Intel MDS. This was necessary in order to run Intel's FORTRAN 77 which was the only available FORTRAN with support for the 8087 floating point math co-processor. For the first software development on the system we chose to implement a simple equivalent systems version of the F-16 flight dynamics as summarized in Fig. 15. Because the F-16 is highly augmented with a sophisticated stability augmentation system (SAS), the attitude control dynamics were modeled with a simple first order lag.

The F-16 has a high thrust to weight ratio and can sustain vertical flight, so quarternions were employed to integrate Euler angles through all possible flight attitudes. Simple body axis acceleration equations were also employed in order to correctly represent lift, thrust, and drag. Thus, the dynamics correctly represented speed and rate of climb in response to thrust changes and attitude maneuvering.

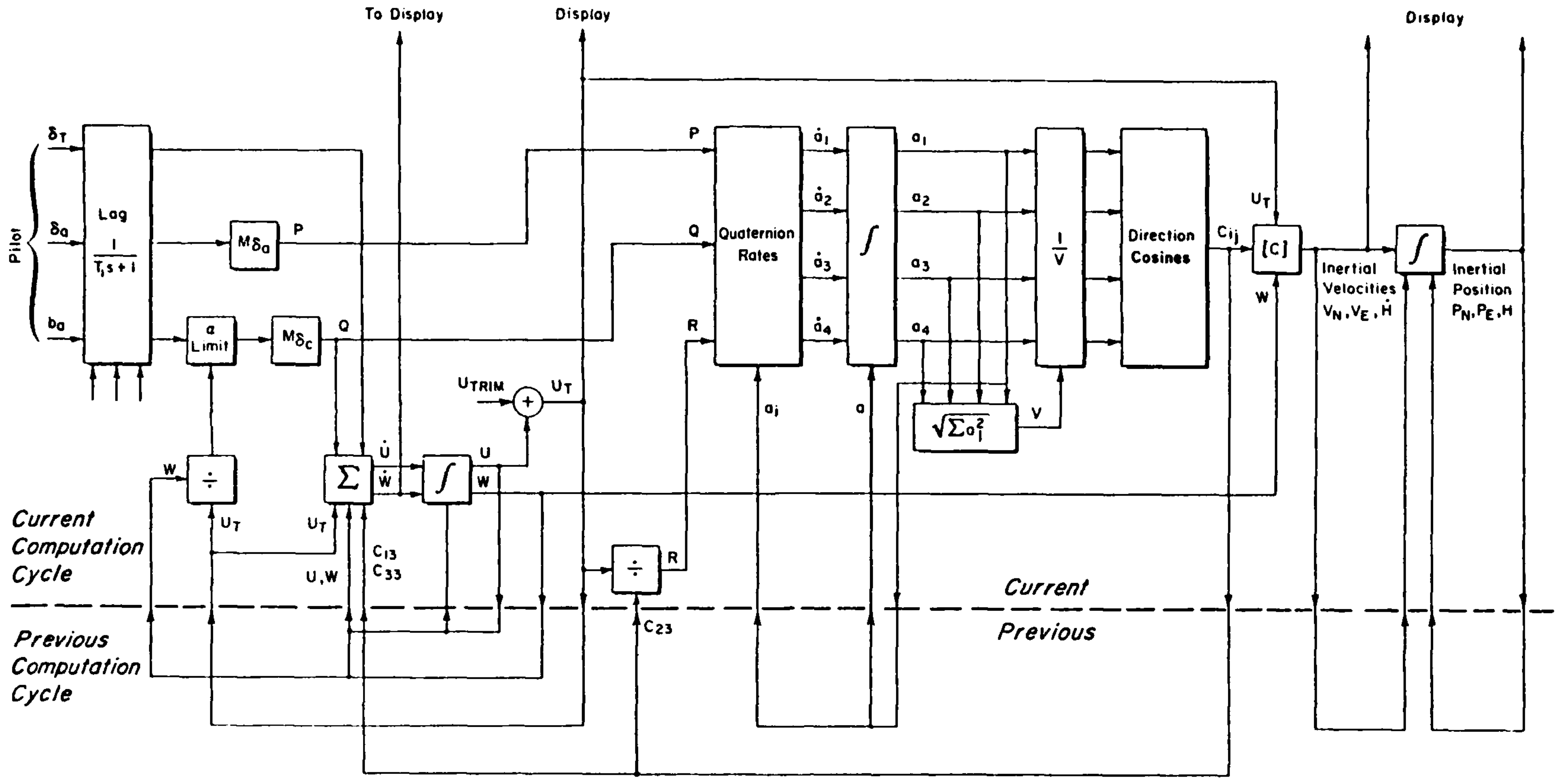


Figure 15. F-16 Dynamics Computation Cycle Data Flow

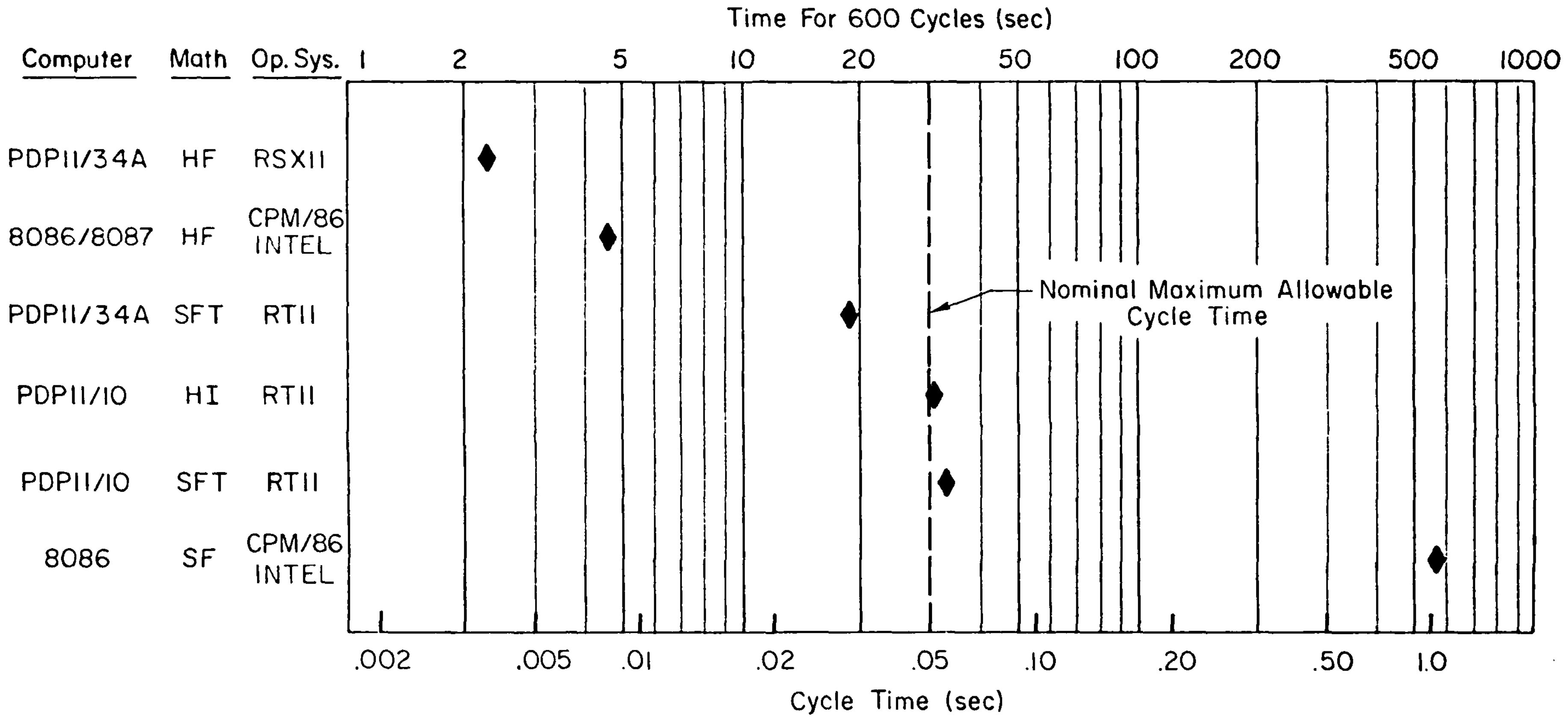
In order to simplify the development and checkout of the flight dynamic equations, they were first tested on a DEC PDP 11/34 system. They were then transferred by disk media to a PDP 11/10 where they were down-loaded to the Intel 8086 system using data communication utilities. Although this process was pursued as a matter of programming expedience it also allowed us to obtain benchmarks on the running speed of the real-time dynamics which were quite impressive.

The benchmarks were obtained by running the dynamics at a 50 msec update rate for 600 iterations which is equivalent of a real-time period of 30 seconds. The actual amount of computational time taken by various computer configurations is shown in Fig. 16. The Intel 8086 without the 8087 (using software evaluation for floating point calculations) required 570 seconds for the equivalent of 30 seconds real-time computation! With the 8087 co-processor the time reduced to 4.6 seconds. Thus the real-time task would be impossible without the 8087 support. Note also in Fig. 15 that the 8086/8087 is about 8 times faster than a PDP 11/10 minicomputer but only a half as fast as a PDP 11/34 minicomputer with floating point math hardware.

The above exercise has demonstrated the 8086/8087 system to be a very powerful processor. The 8086 was originally selected because of the availability of the 8087, and real-time software effort has vindicated this decision. It would appear that the 8086/8087 system is a good general choice for training systems where inexpensive but fast floating point arithmetic is required.

C. INTERFACE SOFTWARE

During real-time operation part of the 8086 host processor's task was to service the various display devices. As discussed in Section III-C, the Atari HUD commands were sent through a parallel data port using a packet protocol. The vector drawing system commands were processed through a general housekeeping module to coordinate the various display components (i.e., ground grid, horizon, attitude bars) then communicated through shared memory with the Z80 processor which controlled the composite data base. The analog angular coordinate system transformations were driven through D/A converters from the 8086.



Math: HF = Hardware Floating Point FPIIA for PDPII/34A or 8C87 for 8086

SFT = Software Threaded

HI = Hardware Integer Non-Threaded With KEIIA

SF = Software

Figure 16. Run Time Benchmarks for Six Degree-of-Freedom Aircraft Dynamics

Checkout, debugging, and real-time tests were just commencing when work on the project was suspended. The basic communication protocols worked without any apparent conflicts. Some work was still required to make the housekeeping functions for the vector display system more efficient in order to meet the real-time update goal of more than 20 interations per second. This work appeared to be straightforward, however, and a matter of developing several assembly language modules which would be linked in with the real-time FORTRAN portion of the host processor code.

SECTION V

SUMMARY AND CONCLUDING REMARKS

The efforts documented in this report, although not carried to completion, clearly demonstrate the feasibility of developing low-cost simulator hardware using commercially available microprocessor elements. Personal computer technology is advancing rapidly, with increasing capability and decreasing cost. Current state-of-the-art microprocessors plus math co-processors exhibit adequate computing power and speed for many applications. Display co-processors also allow relatively complex displays at fast update rates. Indications are that microprocessor performance developments will continue in the near future as 16 and 32 bit applications mature. Display hardware boards designed to operate on personal computer busses will also increase in capability, including the capability for combining video disk and computer generated imagery.

Specific advancements on this project which might find future application include microcomputer-generated HUD and stores control panel displays, a low-cost fast update rate vector drawing processor, and simplified six degree-of-freedom vehicle dynamics. These developments are suitable for implementation with current microprocessor hardware, and would exhibit significantly improved capability with advanced 16 or 32 bit processors. Future applications should take into account the real-time display update requirements discussed in Appendices A, B, and C, however, so as to avoid achieving increased capability in scene complexity at the expense of throughput delays and inadequate update rates.

REFERENCES

1. Flight Manual: USAF/EPAF Series Aircraft F-16A/B, T.O. 1F-16A-1, 30 Jan. 1981.

2. Operational Supplement Technical Manual -- Non-nuclear Munitions Delivery, USAF/EPAF Series F-16A/B Aircraft, T.O. 1F-16A-34-1-15-3, 8 Aug. 1980.

3. F-16 Avionic System Manual, General Dynamics, 1 Sept. 1980, 16PR1342.

4. Stein, Anthony C., R. Wade Allen, and Stephen H. Schwartz, "Use of Reward-Penalty Structures in Human Experimentation," Fourteenth Annual Conference on Manual Control, NASA CP-2060, Nov. 1978, pp. 267-278.

5. Allen, R., Wade, Jeffrey R. Hogge, and Stephen H. Schwartz, "A Simulator for Research in Driver, Vehicle and Environment Interaction," Systems Technology, Inc., P-184, presented at the 56th Annual Meeting of the TRB, Jan. 1977.

6. Garrett, Billy, "Microvec, The Other Type of Video Display," Byte, Vol. 7, No. 11, Nov. 1982, pp. 508-528.

APPENDIX A

COMPUTATIONAL CONSIDERATIONS IN REAL-TIME SIMULATION COMPUTER GRAPHICS

A. OVERVIEW

The display interface is a critical element in the manual control of vehicles. A well designed display device should, at a minimum, not complicate the human operator's control task; in fact, good synthetic displays should augment the operator's control capability. Advances in digital technology and CRTs are causing a revolution in real-time simulation; and general improvements in computer graphics offer a range of computational techniques for creating complex formats.

The problems encountered in digitally generated simulation displays involve computational and refresh update rates, and various effects arising from the quantized nature of digital computations. The human operator requires a display presentation with smooth apparent motion, and without significant delay of visual information feedback. This appendix discusses the nature and implications of various quantization related artifacts in visual display systems.

B. INTRODUCTION

Visual displays are the primary means for providing feedback to the human operator in vehicle control tasks such as car driving and aircraft piloting. Synthetic displays are used to supplement or replace real world cues under conditions of reduced visibility aircraft operations, and in a wide range of simulator applications (i.e., ground vehicle, marine, aircraft, and spacecraft). To be effective, synthetic visual feedback displays should accurately represent the visual cues required by the human operator. Display accuracy applies to both scene content, and temporal considerations associated with intensity, apparent motion, and delays in generating the displayed information.

Electromechanical instruments, and analog simulations including video/terrain model displays, in the past provided acceptable display solutions but lacked flexibility in changing formats and/or creating various visual effects. The advent of computer generated imagery (CGI) has overcome many of the previous limitations, but has added a host of new concerns including computational delay, scene update rate, and quantization effects. Recent advances in display processing architecture and algorithms are largely overcoming basic problems in presenting adequate scene content, but acceptable computational delay is still an unresolved issue.

There is an obvious tradeoff between visual scene complexity, scene update rate and computational delay. An appropriate balance among these three factors is difficult to specify, however, and depends heavily on the specific nature of the human operator's task. In cases where subtle visual cues are important and are near human visual perceptual thresholds, scene content and complexity will be most important. On the other hand, in situations where rapid vehicle maneuvering is required under high bandwidth closed-loop control, computational delay and scene update rate are obviously of great importance. Attention in the literature has been heavily slanted towards considerations of scene realism, encoding of geometric information, spatial orientation, etc. (e.g., Ref. A-1). In this appendix, the temporal aspects of computer generated imagery are emphasized, particularly with regard to the human operator's need for visual feedback in exerting tight, high bandwidth closed-loop control.

C. DISPLAY REQUIREMENTS AND EVALUATION CRITERIA

From the human operator's point of view, a synthetic display should provide a smooth representation of vehicle motions associated with position and angular orientation. Many tasks require the operator to anticipate future motion (commonly referred to as lead generation), so that the display representation should be sufficiently rich and frequent to allow the estimation of the velocity or rate of change of displayed variables. The adequacy of synthetic displays for providing such cues can be evaluated from several points of view.

In real world applications such as displays for vehicle control, we are concerned about the adequacy of system performance and minimizing the potential for human error. For simulator applications the concern relates to the comparability of simulation vs. real-world system experience. For research simulators we wish to have fairly broad, pure comparability. For training applications, we care primarily about the degree to which simulator experience translates to real-world system operation and in some sense makes training safer and more cost effective (e.g., Ref. A-2). For licensing and certification applications, the primary issue is whether a simulator can provide an appropriate criterion or metric for subsequent human operator performance -- a target real-world system. (In some sense a simulator might provide a better test than the actual real-world system because failure modes and incipient accident scenarios might be simulated that would be too hazardous to attempt a real vehicle.)

Two general simulation evaluation criteria that will be considered in this appendix are fidelity and validity. Fidelity relates to the human operator's subjective experience in a simulator, and whether his responses are consistent with the way he would react in the real world. Validity is concerned with more objective realism such as whether engineering system response measurement and performance results are comparable between simulator and the real world.

Simulator fidelity can relate to sensory feedbacks other than display systems (e.g., proprioceptive, motion, and auditory cues) and can interact to a significant degree with visual display fidelity and validity (e.g., the coordination of visual and other cues). The primary purpose of this appendix is to consider CGI display issues, however, and in particular their temporal characteristics. The temporal characteristics to be addressed include refresh rate for avoiding flicker, scene update rate to achieve smooth apparent motion, and overall throughput computational delay which affects system dynamic response (a good current review of temporal effects aside from computational delay can be found in Ref. A-3).

Refresh requirements to minimize flicker are fairly well understood, and are reviewed next for context, because they are sometimes confused with update rate requirements for achieving the illusion of smooth apparent motion. Discussion next turns to apparent motion effects which are quite complex and less well understood than the flicker problem. The update rate requirement for achieving smooth motion is often confused with the computational throughput requirements for minimizing delays in visual feedback information. The computational delay issue, which is the final topic discussed here, has received much engineering attention since the introduction of digital computers into simulation technology, but requirements are still not very well understood.

D. DISPLAY REFRESH RATE AND FLICKER

The steadiness or constancy of display brightness is determined by the sensory characteristics of the retina. Light sources with varying brightness will be perceived as flickering depending on the absolute brightness and relative amount and timing of brightness variations. Much early research was accomplished with flashing light sources and rotating mechanical apparatus which periodically controlled the appearance of light and dark areas (see reviews in Refs. A-4 through A-7).

Over a fairly wide luminance range relevant to visual displays, refresh rates required to avoid flicker are a logarithmic function of display screen brightness as illustrated in Fig. A-1. The eye is generally more sensitive to flicker in the periphery and with increases in the areas of the flickering field. Common video refresh rates, typically 50 or 60 Hz, sometimes appear to flicker when turned up to high intensity, when viewed at close range, and/or when viewed peripherally. Interlaced scanning is frequently used in video presentations to reduce bandwidth and computational requirements. In this case, alternate lines are refreshed once every other scan, which reduces the scan rate of an individual line to 25-30 Hz. Frame rates for ordinary movies run at 24 frames per second, and a shutter is used to interrupt each frame once or twice in addition to the interframe blank interval. Early research found that the critical flicker frequency or CFF varied with several

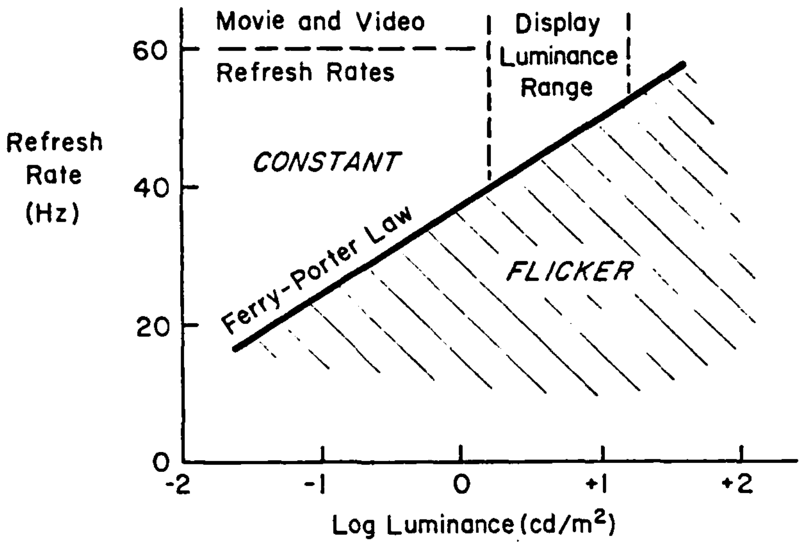


Figure A-1. Effects of Refresh Rate and Image Luminance on Critical Flicker Frequency (i.e., point at which image appears to have constant brightness). Ferry-Porter Law shown for photopic adaptation. Video, movie and CRT strobe written displays commonly set at 60 Hz refresh rate.

parameters associated with the details of the time course of the intensity wave form. Subsequent research and analysis has shown that most of the early empirically observed effects can be explained by a temporal modulation transfer function (Ref. A-8), which is basically a frequency response function for visual sensitivity to time modulated light.

The effects of Flicker are primarily distracting in nature, and can lead to visual discomfort. Such effects are reviewed in Ref. A-6 with regard to design apsects of video display terminals. These effects can easily result in some degradation of a graphics display in terms of fidelity. More important, however, would be the influence of flicker on the perception of subtle motion cues. Flicker can disrupt the pursuit of smooth motion and interact with apparent motion effects discussed next. Peripheral vision is most sensitive to displayed motion, and is also most sensitive to flicker effects. Thus, wide field of view displays will be most sensitive in this regard.

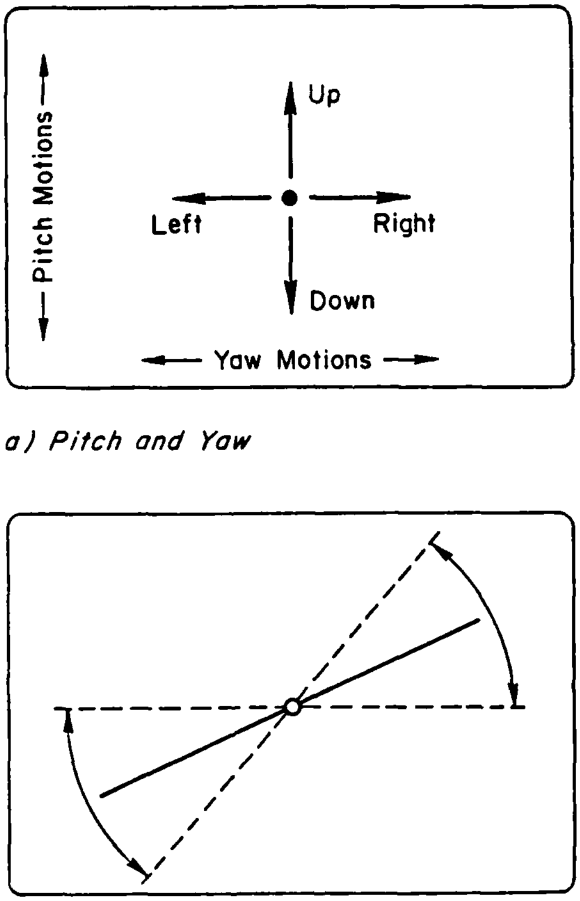
E. SCENE UPDATE RATE AND APPARENT MOTION

Computer generated displays attempt to give the illusion of motion with the presentation of a rapid succession of static scenes or frames. Smooth appearing motion is essential in a vehicle control simulation in order that the human operator can anticipate vehicle movement (i.e., referred to as "lead generation" in manual control terms, Refs. A-9 and A-10). A variety of effects can influence the illusion of smooth motion, including display quantization and a variety of perceptual effects which come under the heading of apparent motion.

The visual system's ability to resolve digital quantization is fairly well characterized by a spatial modulation transfer function (Refs. A-8 and A-11). Classical measures of visual acuity show resolution down to a few minutes of visual arc (Refs. A-4 and A-5), and spatial MTF's typically show spatial bandwidth on the order of 10 cycles per degree (i.e., ~6 minutes of arc/cycle). Therefore, from a strict resolution point of view, display quantization should be on the order of a few minutes of visual arc or less to avoid quantization influence on smooth motion.

Apparent motion in computer generated displays is more than a matter of resolution, however, and in fact, smooth appearing motion can be represented with resolutions much coarser than the visual systems acuity limit. The illusion of apparent motion occurs when two spatially separated stationary images are displayed in rapid succession. The factors influencing the perception of apparent motion involve image frame rate, image intensity, and image displacement between frames. With adequate combinations of these variables, static succeeding images will appear to flow together with smooth motion. Also, apparent size and/or depth effects can be achieved through the control of image brightness. These two types of apparent motion are sometimes referred to as Beta and Gamma movement respectively (Ref. A-12).

Vehicle control simulations require perspective displays of three-dimensional scenes. It is difficult to specify frame update rate and display intensity to meet apparent motion requirements since frame-to-frame image displacement depends on simulated vehicle motion which is under human operator control. Both vehicle (observer) attitude and translation motions influence image displacements. As illustrated in Fig. A-2 vehicle attitude changes in pitch (elevation angle) and yaw (azimuth or heading angle) to a first approximation respectively cause vertical and horizontal displacements of the perspective display plane image. Thus pitch rate and yaw rate combined with the scene update rate to define scene-to-scene image displacement. Since real vehicles have physical limits to achievable pitch and yaw rates under normal conditions, display characteristics could be specified accordingly.



a) Pitch and Yaw

b) Roll (Rotation)

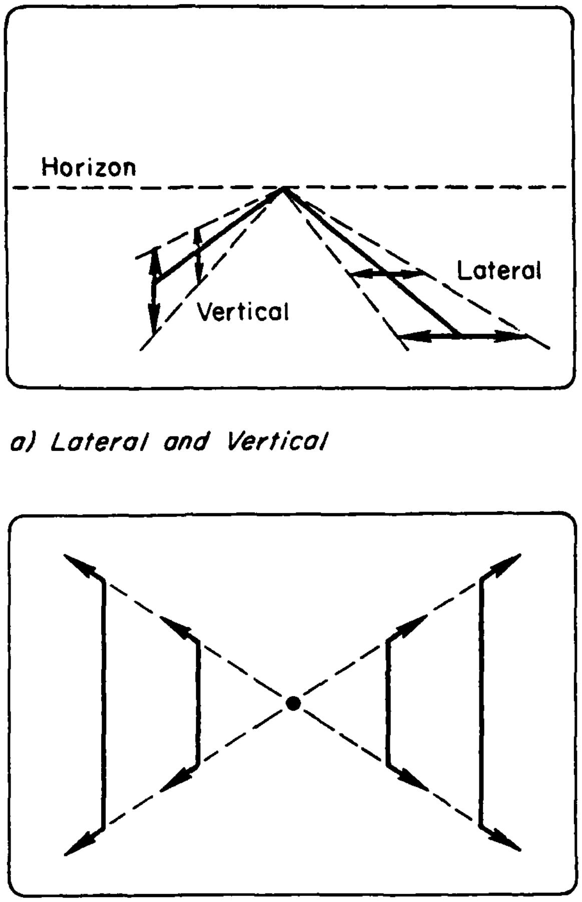
Figure A-2. Effects of Vehicle Angular Orientation on Perspective Display Plane Image Motion

Changes in vehicle roll angle require the display to rotate about some point in the perspective display plane. Image velocity is highest at the display edge in this case, and maximum roll rates can be defined for a given vehicle relative to specifying display requirements. The horizontal scan lines of raster scan displays present a particular problem for rotating images. Near horizontal lines can have a staircase appearance which can create both static and dynamic visual cue artifacts. Antialiasing algorithms can make significant improvements to this situation.

Perspective plane image motions resulting from vehicle translation are illustrated in Fig. A-3. Velocity parallel to the display plane causes scene elements closesr to the observer to displace more rapidly than more distant score elements. Because of characteristics of vehicle dynamics and motion kinematics, the image motions caused by lateral and vertical vehicle motions are usually small and would not tend to influence the specification of scene update rate.

The effect of vehicle longitudinal (forward) velocity can have a significant impact on scene update rate, however. Velocity normal to the display plane causes scene elements to have increasing angular velocity as they move towards the edge of the display. It can be shown geometrically that this is not a matter of absolute velocity, but of velocity relative to the range of an object. Thus, ground vehicles and helicopters can generate just as high scene expansion rates as high speed aircraft because they typically operate in closer proximity to scene elements.

The several types of vehicle-generated scene motion described above combine to determine frame-to-frame scene displacements. Scene update rate requirements depend on providing adequate visual control cues to the human operator, and avoiding distracting visual artifacts. A large part of the visual information required by a human operator comes from the region where the vehicle velocity vectors converge at infinite range (the focus of expansion). Thus, while some frame-to-frame stepping motion in the periphery is probably may be tolerated, the limiting parameters have not been firmly established.



a) Lateral and Vertical

b) Longitudinal

Figure A-3. Effects of Vehicle Translational Motion on Perspective Display Plane Image Motion

Given that adequate smooth scene motion is provided to the human operator, it is likely that appropriate visual motion cues will be perceived. Another critical factor still needs to be considered, however, and that is the degree to which the feedback of this information has been delayed by computational processing. Rapid scene update rates that achieve smooth apparent scene motion do not necessarily imply that the feedback information was supplied to the human operator fast enough. (Some digital display processors require multiple frame intervals for input information to work its way through to image motions.) The influence of the speed of information feedback on system response is based on manual control theory which is the final discussion point in this appendix.

F. COMPUTATIONAL DELAY AND SYSTEM DYNAMIC RESPONSE

In current simulator mechanizations, vehicle equations of motion are typically solved on a digital computer, which then drives a digital graphics system. This architecture leads to two sources of computational delay which may total well in excess of 100 msec. The effects of computational delay on simulator performance has been a continuing source of concern (Refs. A-13 and A-14), but guidelines for tolerable levels of delay have been elusive.

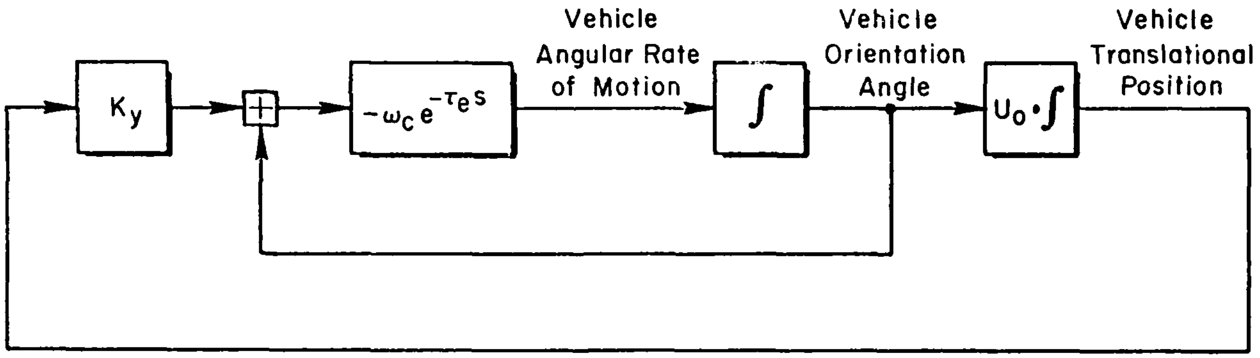
The basic problem with computational delay lies in its effect on the human operator's closed-loop control of vehicle motions. A simplified model of a manual control system is illustrated in Fig. A-4. This model could represent the dynamics of an automobile driver's steering task (Ref. A-15), or even a rough approximation of a pilot/aircraft tracking a second aircraft (Ref. A-16). As discussed in Ref. A-15, the bandwidth of this system, ω_c, is dictated by the effective time delay, τ_e. The effective time delay in turn is a composite effect of the human operator's characteristics and vehicle dynamics (the human operator compensates for the vehicle dynamics to a certain degree, and the composite dynamics can be crudely modeled as a time delay).

As derived in Ref. A-15 the human operator can increase the system bandwidth up to some basic stability limit. This relationship is given in Fig. A-4, which is composed of the normalized bandwidth product, $\omega_c\tau_e$, and another composite factor associated with the control weighting the operator places on vehicle translation, K_y, and forward velocity, U_o. Review of several manual control research studies (Ref. A-17) has indicated that human operators typically maintain a bandwidth which allows some finite stability margin (phase margin in control theory parlance). As system equivalent time delay increases in going from real vehicles to fixed base simulators, human operators maintain a consistent stability margin by reducing system bandwidth. This effect is illustrated in Fig. A-5.

ω_c = Operator / Vehicle Control Bandwidth

τ_e = Effective Time Delay (operator + vehicle and computational delays)

K_y = Translation Feedback Gain


K_y
$-\omega_c e^{-\tau_e s}$
Vehicle Angular Rate of Motion
$\int$
Vehicle Orientation Angle
$U_o \cdot \int$
Vehicle Translational Position

Stability Criteria : $\tau_e \omega_c + \frac{K_y U_o}{\omega_c} < \frac{\pi}{2}$

Figure A-4. Simplified Crossover Model for Manual Vehicle Control and Stability Requirements (adapted from Ref. A-15)

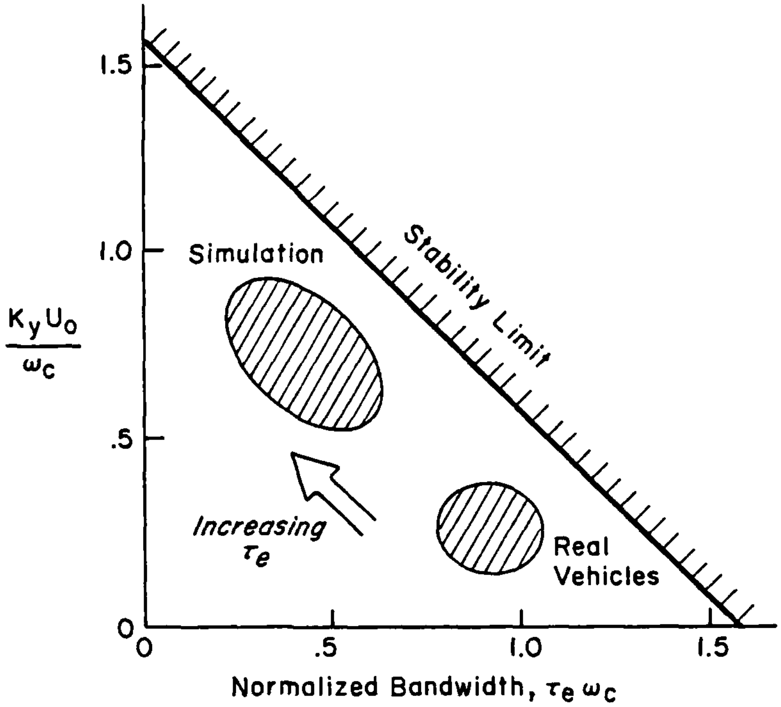


Figure A-5. Effect of Increased Effective Time Delay on System Bandwidth (Adapted from Ref. A-15)

Presumably, the addition of computational delay would cause decreased bandwidth in the human operator's response which would certainly have consequences in system response and performance. In the context of aircraft handling qualities, pilot opinion rating has also been shown to degrade with an increase in equivalent system time delay (Ref. A-18). Further research is still required to determine all of the implications of computational delays and their interaction with other system characteristics. Some previous work has indicated that the human operator compensates for time delays (Ref. A-19). Other work suggests that computer system compensation can also be used to partially conteract some of the effects of computational delays (Ref. A-20).

Further research and analysis is required in order to completely understand the above relationships. It should be noted, however, that human operator compensation for computationally induced time delays in simulators represents a change in behavior from real world operations thus presenting a fidelity issue. Reduced system bandwidth due to added time delay will adversely affect system performance over real-world operations which can, in turn, degrade effect simulator validity.

A comprehensive study of the dynamics of computational delays in simulators should be conducted so that display manufacturers will have realistic performance criteria available when making tradeoffs between scene complexity and computational delay. Also, the requirements of a manual control system may help define optimum architectures for computer display processors. For example, referring to Fig. A-4, it can be shown that delays in feedback of angular motion are much more serious than delays in translational motion which are one integration further removed from the human operator's control actions. This suggests that angular transformations need to be updated most frequently, and perhaps translational transformations could be updated less frequently to achieve some savings in computation time.

G. CONCLUDING REMARKS

As reviewed in this appendix, there are a variety of human operator characteristics that must be taken into account when setting requirements for computer generated imagery. The temporal requirements reviewed here relate to achieving smooth image motion and minimizing computation delay. Further research is required in aspects of apparent motion and the effects of computational delay before specific requirements can be stated. There are potential tradeoffs which might be made in display processor architecture and scene complexity vs. update rate once the human operator's requirements are more completely understood.

APPENDIX A

REFERENCES

A-1. Richards, Whitman, and Key Dismukes, Vision Research for Flight Simulation, AFHRL TR-82-6, July 1982.

A-2. Semple, Clarence A., Simulator Training Requirements and Effectiveness Study (STRES): Executive Summary, AFHRL TR-80-63, Jan. 1981.

A-3. Sekuler, R., P. D. Tynan, and R. S. Kennedy, Sourcebook of Temporal Factors Affecting Information Transfer from Visual Displays, Army Research Inst. for Behavioral and Social Sciences, Technical Report 540, June 1981.

A-4. Woodworth, Robert S., and Harold Schlosberg, Experimental Psychology, Revised Edition, Holt, Rinehart and Winston, New York, Jan. 1964.

A-5. Stevens, S. S. (Ed.), Handbook of Experimental Psychology, John Wiley and Sons, Inc., New York, 1951.

A-6. Panel on Impact of Video Viewing on Vision of Workers, National Research Council, Video Displays, Work and Vision, National Academy Press, Washington, D.C., 1983.

A-7. Kraiss, K. F., and J. Moraal (Eds.), Introduction to Human Engineering, Verlag TUV Rheinland GmbH, Germany, 1976.

A-8. Cornsweet, Tom N., Visual Perception, Academic Press, San Francisco, CA, 1970.

A-9. McRuer, D. T., and E. S. Krendel, Mathematical Models of Human Pilot Behavior, AGARDograph No. 188, Jan. 1974.

A-10. McRuer, Duane, "Human Dynamics in Man-Machine Systems," Automatica, Vol. 16, No. 3, May 1980, pp. 237-253.

A-11. Ginsburg, Arthur P., Visual Information Processing Based on Spatial Filters Constrained by Biological Data, AMRL-TR-78-129, Vols. I & II, Dec. 1978.

A-12. Murch, Gerald M., Visual Auditory Perception, Bobbs-Merrill Co., Inc., Indianapolis, IN, 1973.

A-13. Queijo, M. J., and Donald R. Riley, Fixed-Base Simulator Study of the Effect of Time Delays in Visual Cues on Pilot Tracking Performance, NASA TN D-8001, Oct. 1975.

A-14. Crane, D. Francis, "Time Delays in Flight Simulator Visual Displays," Proceedings of the 1980 Summer Computer Simulation Conference, Seattle, WA, pp. 552-557.

A-15. Allen, R. Wade, "Stability and Performance Analysis of Automobile Driver Steering Control," SAE Paper 820303, presented at the International Congress and Exposition, Detroit, MI, 22-26 Feb. 1982.

A-16. Hoh, Roger H., Thomas T. Myers, Irving L. Ashkenas, et al., Development of Handling Quality Criteria for Aircraft with Independent Control of Six Degrees of Freedom, AFWAL-TR-81-3027, Apr. 1981.

A-17. Allen, R. Wade, "Modeling Driver Steering Control Behavior," Systems Technology, Inc., P-322, presented at the 1982 IEEE International Conference on Cybernetics and Society, Seattle, WA, Oct. 1982.

A-18. Hodgkinson, J., W. J. LaManna, and J. L. Heyde, "Handling Qualities of Aircraft with Stability and Control Augmentation Systems -- A Fundamental Approach," The Aeronautical Journal, Vol. 80, No. 782, Feb. 1976, pp. 75-81.

A-19. Hess, Ronald A., "The Effects of Time Delays on Systems Subject to Manual Control," AIAA Paper 82-1523, AIAA Guidance and Control Conference Proceedings, San Diego, CA, 9-11 Aug. 1982, pp. 165-172.

A-20. Gum, D. R., and W. B. Albery, "Time-Delay Problems Encountered in Integrating the Advanced Simulator for Undergraduate Pilot Training," J. Aircraft, Vol. 14, No. 4, Apr. 1977.

APPENDIX B

EFFECTS OF TRANSPORT DELAYS ON MANUAL CONTROL SYSTEM PERFORMANCE

A. OVERVIEW

Throughput or transport delays in manual control systems can cause degraded performance and lead to potentially unstable operation. With the expanding use of digital processors, throughput delays can occur in manual control systems in a variety of ways such as in digital flight control systems in real aircraft, and in equation-of-motion computers and CGI's in simulators. Previous research has shown the degrading effect of throughput delays on subjective opinion and system performance and dynamic response. A generic manual control system model is used in this appendix to provide a relatively simple analysis of, and explanation for, the effects of various types of delays. The consequences of throughput delays of some simple system architectures is also d cussed.

B. OVERVIEW AND BACKGROUND

Past literature surveys associated with flight simulation fidelity have found that system response lags and computational delays cause performance and pilot subjective rating problems (Refs. B-1 and B-2). Pilot/vehicle model analysis has shown that delays on the order of 50 to 100 msec can have an appreciable influence on performance and workload (Ref. B-3). Recent experiments have shown performance effects of time delays which are consistent with model analysis (Refs. B-4 and B-5).

The above literature indicates that simulator computational delays can have a serious effect on aircraft simulation fidelity. Ground vehicles typically have faster response dynamics than aircraft in terms of path control, and it is suspected that the problem may be even more serious for driving simulators. To further understand the effect of various potential sources of transport delays a computer model analysis was undertaken using a generic vehicle control model as described below.

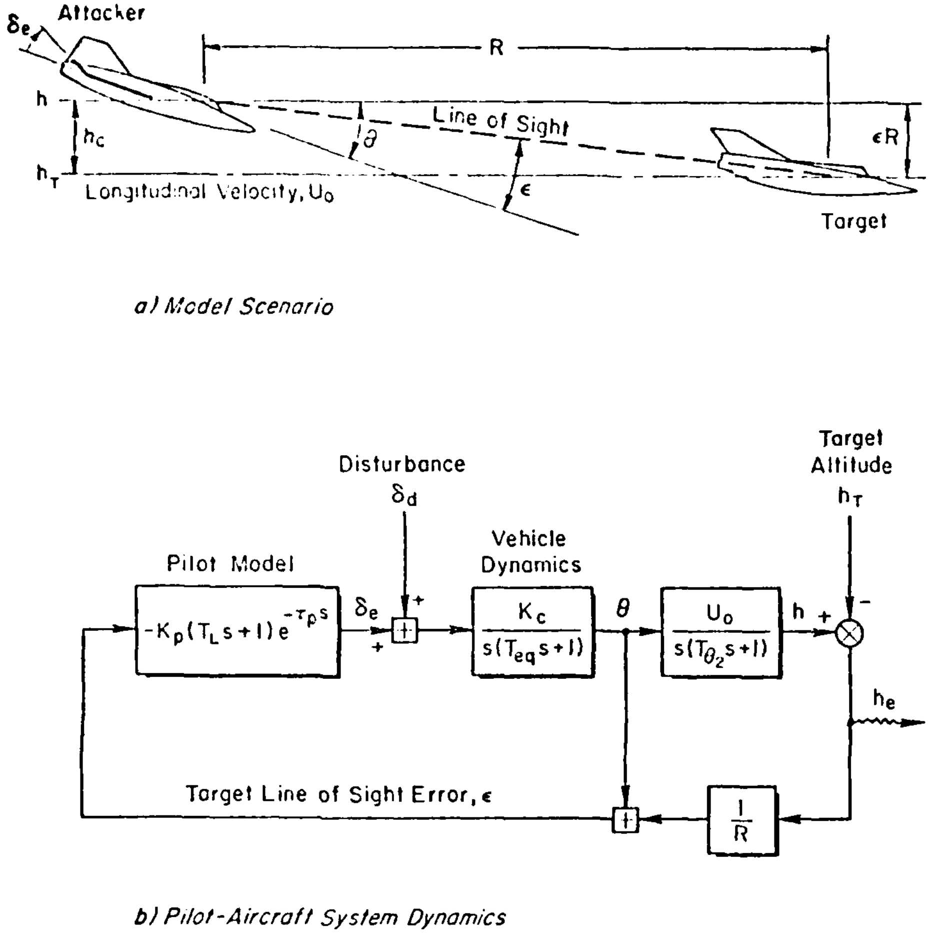
The analysis was carried out to study the effect of several sources of computational delay including host computer system, display system, and motion system. (This analysis does not address another important simulation artifact, that of the mismatch between visual and motion cues, which can lead to vertigo and/or sickness.)

C. ANALYSIS MODEL

The basic control example for the analysis model concerns generic vehicle tracking (e.g., dogfighting) where the operator must point his vehicle at a target or aim point at some fixed distance in front of the vehicle. An example for a typical aircraft is shown in Fig. B-1 (Ref. B-6). A similar arrangement holds for ground vehicle steering control as illustrated in Fig. B-2 (Ref. B-7). The only dynamic difference between the car and aircraft examples is the T_{θ_2} path lag which is ignored for the car. (It actually exists in the car, but as a very high frequency lag corresponding to an aircraft with steep lift or side-force curve slopes.)

A generic operator/vehicle pointing control model was prepared for analysis based on an expansion of the Figs. B-1 and B-2 models. A block diagram of the analysis model is shown in Fig. B-3 which has additional dynamic complexity over the simplified models of Figs. B-1 and B-2 as follows:

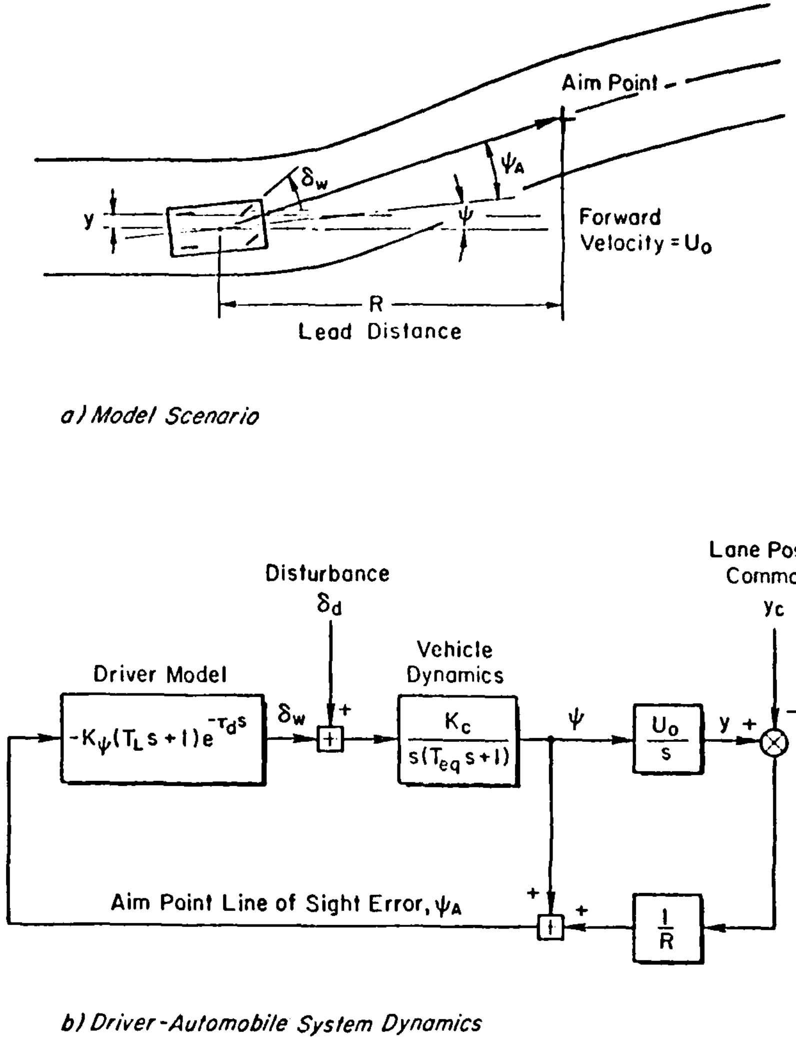
- Pilot lead generation to compensate for effective vehicle lag, T_{eq}, is provided by angular rate feedback which is assumed to represent a composite of motion perception (i.e., acceleration, angular rotation and proprioceptive sensations).
- Lightly damped, second-order limb/manipulator dynamics.
- Human operator transport delay associated with visual (τ_v) and motion (τ_r) perception.
- System transport delays associated with dynamic computations (τ_c), display generation (τ_d), and motion feedback (τ_m).



a) Model Scenario

b) Pilot-Aircraft System Dynamics

Figure B-1. System Model for Air-to-Air Target Tracking



a) Model Scenario

b) Driver-Automobile System Dynamics

Figure B-2. System Model for Car/Driver Path Following a Commanded Path

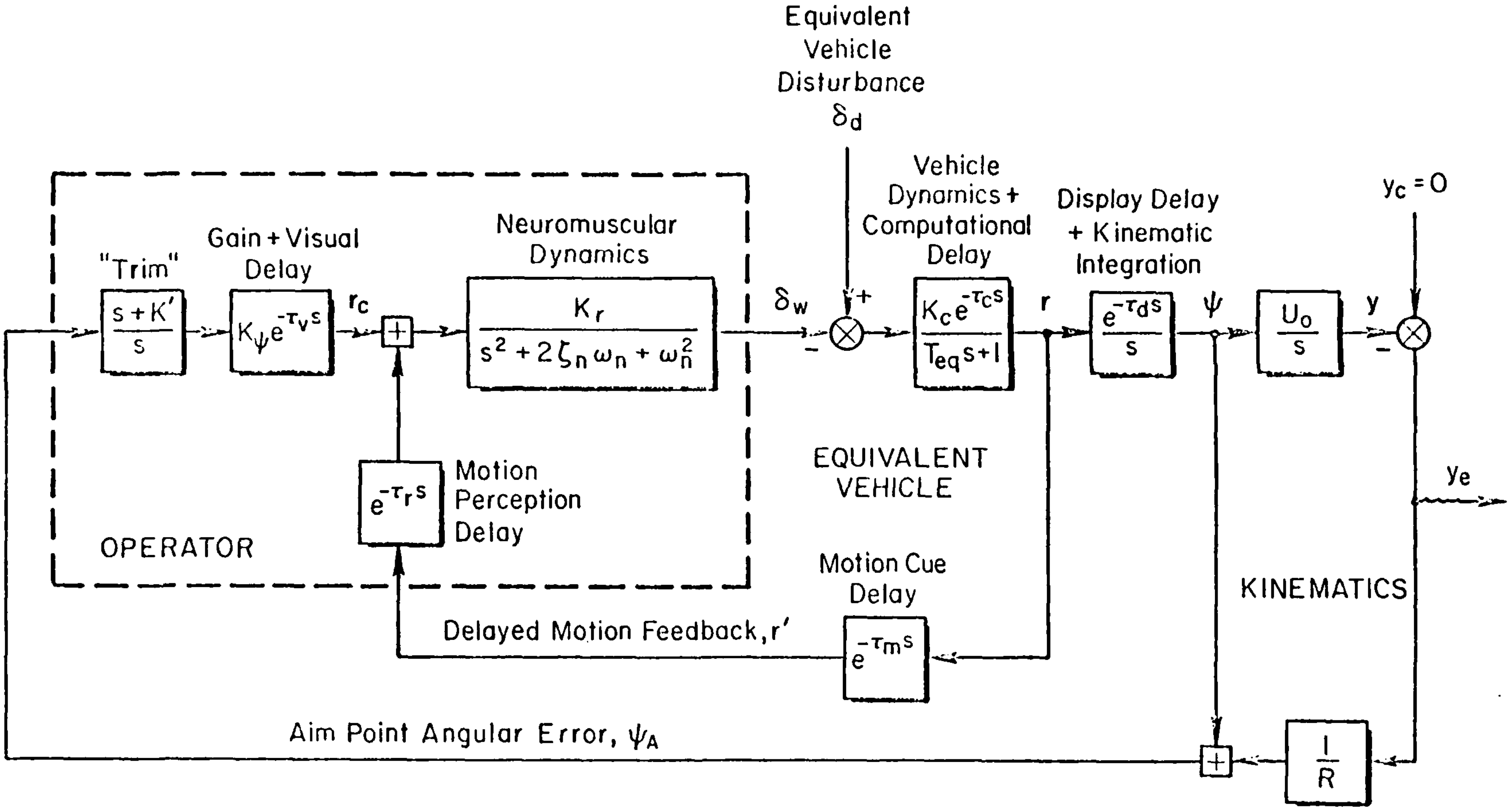

Equivalent Vehicle Disturbance
δ_d
"Trim"
$\frac{s+K'}{s}$
Gain + Visual Delay
$K_\psi e^{-\tau_v s}$
r_c
Neuromuscular Dynamics
$\frac{K_r}{s^2+2\zeta_n\omega_n+\omega_n^2}$
δ_w
+
−
Vehicle Dynamics + Computational Delay
$\frac{K_c e^{-\tau_c s}}{T_{eq}s+1}$
r
Display Delay + Kinematic Integration
$\frac{e^{-\tau_d s}}{s}$
ψ
$\frac{U_o}{s}$
y
$y_c = 0$
−
y_e
Motion Perception Delay
$e^{-\tau_r s}$
OPERATOR
EQUIVALENT VEHICLE
Motion Cue Delay
$e^{-\tau_m s}$
Delayed Motion Feedback, r'
KINEMATICS
Aim Point Angular Error, ψ_A
$\frac{1}{R}$

Figure B-3. Generic Operator/Vehicle Tracking Dynamic Model for Analyzing Transport Delay Effects

- A low frequency trimming operation to minimize low frequency "hang off" errors.

In the Fig. B-3 analysis model, a disturbance (δ_d) is added at the input to the equivalent vehicle dynamics to represent the effects of wind gusts, and roadway inputs in the case of ground vehicles. The equivalent vehicle dynamics are represented by s simple first-order time constant, T_{eq}, to approximate lags in vehicle rotational rate in response to control inputs. Path lag, T_{θ_2}, is assumed to be zero for this analysis. Transport delay representations are defined below.

D. TRANSPORT DELAY SOURCES

The model analysis was arranged to assess the effects of three sources of computational delay. The first is a transport delay associated with the vehicle dynamics equations of motion (τ_c). This delay could represent the equivalent delay used in specifying vehicle handling qualities (Ref. B-8) which can result from the composite effect of stick filters, digital flight control system delays, and control system and other high frequency vehicle dynamics effects. It could also represent the composite effect of A/D and D/A sampling holds, integration routines and computational cycle time. The analysis considered either no delay, which might correspond to an analog vehicle or an analog simulation computer, or a delay of 0.075 sec, which is a common equivalent delay time associated with complicated digital simulation computations or modern high performance aircraft with digital flight control systems.

The second delay source considered was that due to display system characteristics. Analysis conditions included either no delay, which might be associated with an analog processor, or 100 msec delay which is common to many of the current generation simulation CGI raster scan devices. The delay time condition might also be associated with the camera servos on a terrain board system, or digital processing in HUD or EADI instruments.

The final delay factor was concerned with motion feedback to the human operator. Analysis conditions included no delay, or a rather long delay of 250 msec. The long-delay condition might be associated with a fixed-based simulator environment where there were no motion cues available, and the human operator has to generate heading rate cues visually. This could also result from motion lags in a simulator motion system combined with computational delay in generating the motion base drive commands. The additional 250 msec was calculated to give model behavior that was consistent with past measurements made under both fixed-based and moving base conditions (Ref. B-9), and is also consistent with delays identified in flight simulators (Ref. B-10).

E. MODEL PARAMETER SELECTION

The Fig. B-3 model has a variety of parameters that must be set to represent either vehicle characteristics or human operator behavior. A nominal vehicle heading time constant (T_{eq}) of about 0.2 sec was selected. This might represent a light weight, high performance aircraft, or a compact to intermediate size automobile. The vehicle gain is somewhat arbitrary, depending both on control gain and vehicle stability derivatives.

The human operator model parameters can be divided into two groups; those which are relatively fixed and were assumed to be constant for this analysis, and other parameters which the human operator typically adapts in order to achieve stable and desirable closed-loop performance. The trim constant (K') was assumed to be constant at 0.5 rad/sec which is consistent with driver measurements discussed in Ref. B-7. The visual time delay (τ_v) was assumed to be constant at 0.05 sec. The time delay associated with motion feedback perception (τ_r) was also set at 0.05 sec. The second-order limb/manipulator system dynamics were set at a break frequency of 20 rad/sec and a damping ratio of 0.5. The pure delay and lag characteristic were set to give a composite effective time delay, with the motion feedback loop closed, of 0.17 seconds which is consistent with past car-driver measurements (Ref. B-7).

The human operator can arbitrarily adapt his inner and outer loop gains (K_r and K_ψ respectively) and has some control over aim point range, R, to optimize system performance and control stability. For the model structure assumed here, K_r was adjusted to obtain as wide a frequency response as possible in the motion feedback loop while maintaining a reasonable closed-loop damping ratio (i.e., $\zeta_{CL} \cong 0.5$). For a real vehicle without any computer delay or extra motion feedback delays the variable K_r would be adjusted to cancel out the effects of the vehicle equivalent heading lag, T_{eq}. As computational delay is added or the heading rate feedback delay is changed, K_r would then be adjusted to still achieve as wide a bandwidth as possible with this inner loop.

When K_r is properly adjusted a fairly flat closed-loop amplitude ratio can be achieved for the motion feedback loop as illustrated in Fig. B-4. When the conditions in Fig. B-4 are achieved the closed-loop response of the motion feedback loop can then be approximated by a gain and an equivalent time delay up to the point where the amplitude ratio begins to roll off:

$$\text{Motion Feedback Closed-Loop Response} \cong K_{eq}e^{-\tau_o}$$

Closed-loop equivalent parameters are given in Table B-1 for the Fig. B-4 response functions.

Note that when there are no extra computational delays and a low feedback delay, as in the upper lefthand corner of Fig. B-4, the closed-loop bandwidth of the heading rate loop can be adjusted to be quite high. Theoretically, in this case the bandwidth is on the order of 15 rad/sec, and the equivalent time delay is quite small (about 120 msec). If τ_o is added to the visual time delay (τ_v), the result is an overall equivalent time delay for the driver of about 0.17 sec, which is consistent with measurements discussed in Ref. B-7. On the other hand, when a significant amount of delay is put into the motion feedback loop, as in the lower righthand corner of Fig. B-4, the closed-loop bandwidth of the heading rate loop is reduced considerably. In this case it is reduced to the vicinity of the vehicle's heading rate time constant (i.e., delayed feedback effectively opens the loop). In the second case the equivalent time delay for the heading rate loop is increased to about 235 msec.

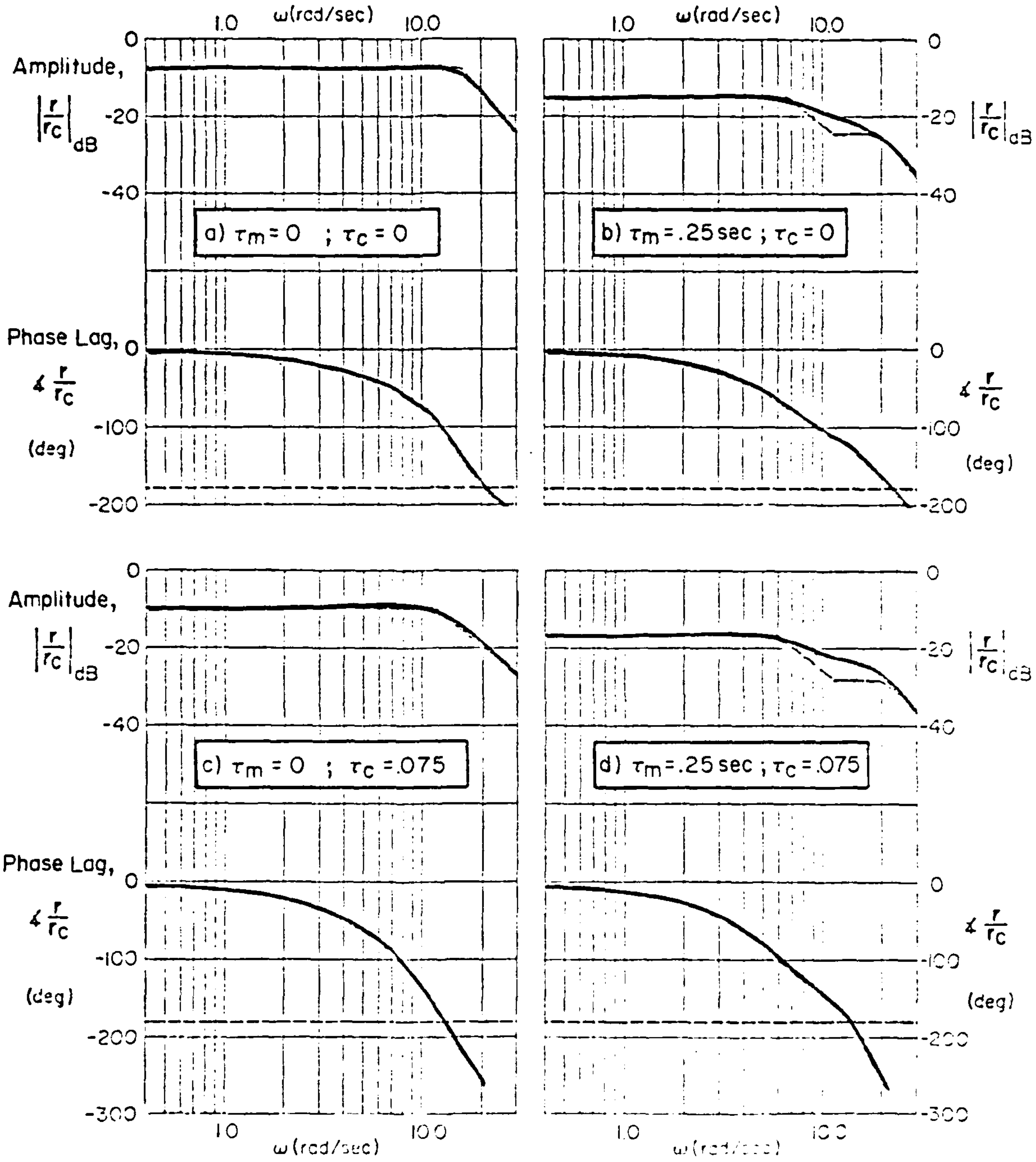


Figure B-4. Motion Feedback Closed-Loop Response Functions (Equivalent Closed-Loop Parameters Given in Table 1)

TABLE B-1. MOTION FEEDBACK LOOP PARAMETERS FOR VARIOUS LEVELS OF MOTION FEEDBACK ($\rightarrow_m$) AND COMPUTATIONAL ($\rightarrow_c$) DELAY

OPEN LOOP			EQUIVALENT CLOSED	
τ_m (sec)	τ_c (sec)	K_r (sec^{-1})	K_{eq} (sec)	τ_o (sec)
0	0	500	0.414	0.12
	0.075	350	0.331	0.20
0.25	0	150	0.175	0.16
	0.075	125	0.150	0.235

F. EQUIVALENT OPERATOR/VEHICLE TIME DELAY EFFECTS

The equivalent closed-loop time delays that are achieved over a wide range of motion feedback delays (τ_m) and two levels of computational delay are illustrated in Fig. B-5. Here, note that the computer computation delay (τ_c) has a much greater influence on the equivalent closed-loop delay than does the motion feedback time delay which is actually in the feedback of this loop. These induced delays will have two effects on human operator/vehicle performance. First, the increased equivalent closed-loop time delay will affect the operator's ability to achieve an overall bandwidth in controlling outer loop errors. Second, the effect of disturbances that act on the vehicle will be delayed in their feedback to the operator. Thus, there will be an overall delay in the human operator responding to a disturbance, and, once the operator responds, he will be limited in the bandwidth of his response.

The parameters that remain to be selected in the Fig. B-3 model are K_ψ and U_o/R. Procedures for optimizing human operator performance by the selection of these two variables has been discussed for car driving in Ref. B-7. The procedure involves breaking the Fig. B-3 model loop at the r_c point and then considering the composite driver/vehicle open-loop transfer function proceeding around the loop.

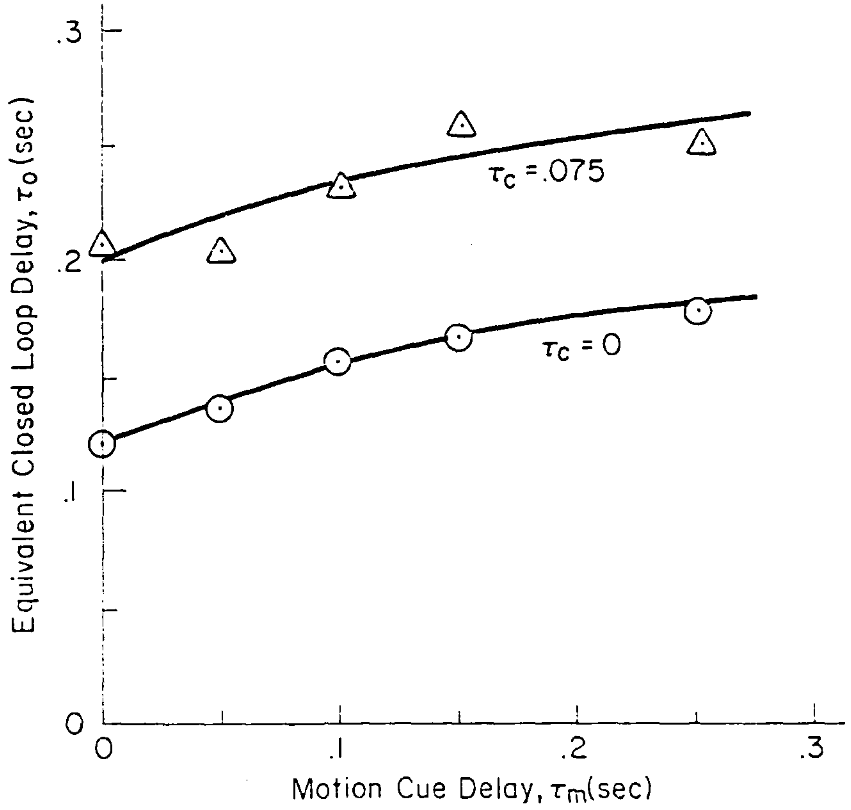


Figure B-5. Equivalent Motion Feedback Delay for Various Levels of τ_m and τ_c

Given that the inner loop closed-loop dynamics can be interpreted as an equivalent time delay over the outer loop bandwidth, then an Extended Crossover Model describing function for the Fig. B-3 model can be written as:

$$Y_p{}^*Y_c = \underbrace{\frac{s + K'}{s}}_{\text{Low Frequency Trimming}} \cdot \underbrace{\frac{s + U_0/R}{s}}_{\text{Low Frequency Kinematic Lead + Integration}} \cdot \underbrace{\frac{\omega_c e^{-\tau_e s}}{s}}_{\text{Crossover Model}}$$

The kinematic zero at U_o/R is at low enough frequency that the dynamics become K/s-like in the region of magnitude crossover (the classical crossover model law). Now the optimum K_ψ and U_o/R values can be interpreted in terms of crossover frequency and phase margin.

The Y_p*Y_c transfer function is illustrated in Fig. B-6 for each combination of induced time delays under consideration. As noted in Fig. B-6, the low frequency effects of aim point kinematics $(s + (U_o/R))/s$ plus trimming $(s + K')/s$ have resulted in a conditionally stable system. The variable U_o/R which corresponds to lead distance or look-ahead range for the human operator's aim point was adjusted to give the stable phase region indicated in Fig. B-6. As can be noted, U_o/R was varied for each combination of the various time delays in Fig. B-6 in order to get a similar stable phase region for all conditions. Once this form had been achieved, then the remaining variable K_ψ was selected in order to give a specified phase margin. The low frequency kinematic and trim effects cause a significant reduction in phase margin in the crossover frequency region and cannot be neglected for tasks requiring control to aim points with speed-to-range ratios in the region of 0.1-1.0 rad/sec. It should be noted that situations which constrain thelook ahead distance R to small values (e.g., driving in the fog, pointing at shorc range ground or air targets) could decrease the region over which the phase is stable.

Phase margin has been used previously as a metric for quantifying the stability of car/driver closed-loop steering performance (Ref. B-11). K_ψ is set to achieve a desired phase margin at the crossover frequency which can be considered the bandwidth of the closed-loop operator/vehicle control system. The phase margin quantifies the stability or oscillatory nature of the operator's steering control behavior. The bandwidth or crossover frequency defines how rapidly the control can be carried out. For this analysis an attempt was made to maintain a constant phase margin of 30 deg for all cases. This level has been typically found in past car driving s.udies (Ref. B-7). The achievable crossover frequency depends on the total system time delay

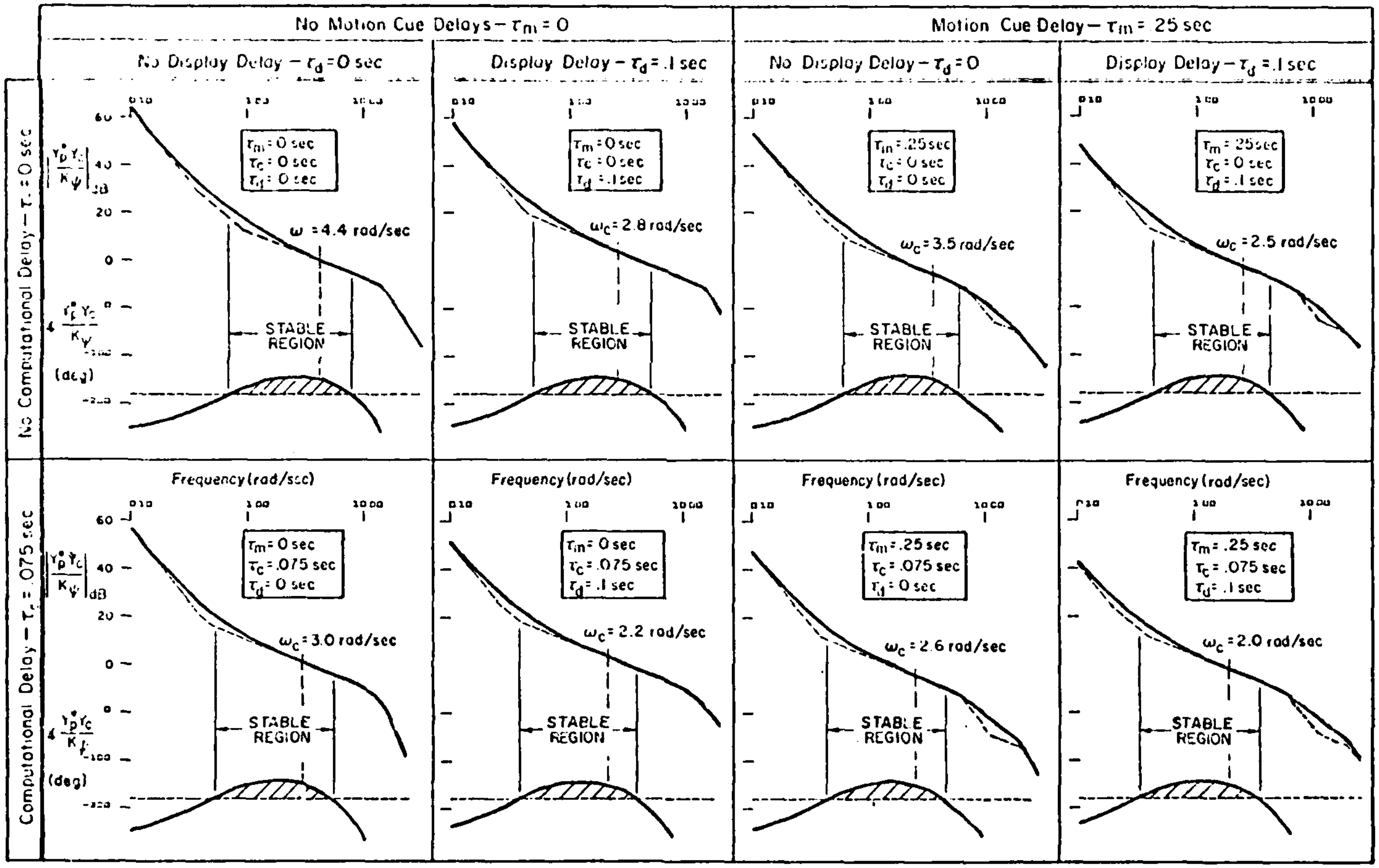


Figure B-6. Equivalent Open-Loop Human Operator/Vehicle Describing Functions For Various Levels of Simulation Time Delays which includes the inner loop equivalent time delay, visual perceptual delay, and display system transport delay:

$$\tau_e = \tau_o + \tau_v + \tau_d$$

Gain and crossover model parameters are summarized in Table B-2.

G. BANDWIDTH EFFECTS

The consequences of the above adjustment procedures can be seen in Fig. B-7. Here observe that the control bandwidth of the operator/vehicle system drops dramatically as various delays are added into the simulation loop. Adding the 0.1 sec display delay has the largest single impact on equivalent time delay and system bandwidth. Motion cue delays had the least impact. Computational delays had an effect somewhere in between motion cue delays and display delays. Perhaps if the computational delay had been 100 msec it would have had a similar effect to the display delay. The concatenation of these various delay sources deteriorates the system bandwidth to an even greater degree. When all the delay sources were combined, the system bandwidth was cut by more than 50 percent.

The relationship shown in Fig. B-7 is a consequence of maintaining a constant phase margin. If we had changed the desired phase margin, or chosen a different aim point range (thus changing the low frequency kinematic root U_o/R) then a different constant would have resulted. In any case, we can use the hyperbolic relationship between ω_c and τ_e to determine how changes in effective system time delay affect achievable bandwidth. Assume that a 25 percent decrease in system bandwidth is permissible. Then

$$\frac{\omega_c'}{\omega_c} = 0.75 = \frac{K/\tau_e'}{K/\tau_e} \rightarrow \tau_e' = \frac{\tau_e}{0.75}$$

or

$$\tau_e' - \tau_e = \Delta\tau_e = \frac{1}{3}\,\tau_e$$

TABLE B-2. HUMAN OPERATOR/VEHICLE GAIN AND CROSSOVER MODEL PARAMETERS FOR VARIOUS COMBINATIONS OF INDUCED VEHICLE/SIMULATOR DELAYS

VEHICLE/SIMULATOR INDUCED DELAYS			GAINS		CROSSOVER MODEL PARAMETERS	
τ_m (sec)	τ_c (sec)	τ_d (sec)	U_o/R (rad/sec)	K_ψ (sec^{-1})	ω_c (rad/sec)	τ_e (sec)
0	0	0	0.92	10.26	4.4	0.17
		0.1 sec	0.44	6.59	2.8	0.27
	0.075 sec	0	0.50	8.60	3.0	0.25
		0.1 sec	0.26	6.39	2.2	0.345
0.25 sec	0	0	0.65	18.51	3.5	0.215
		0.1 sec	0.35	13.47	2.5	0.305
	0.075 sec	0	0.38	16.13	2.6	0.29
		0.1 sec	0.20	12.58	2.0	0.38

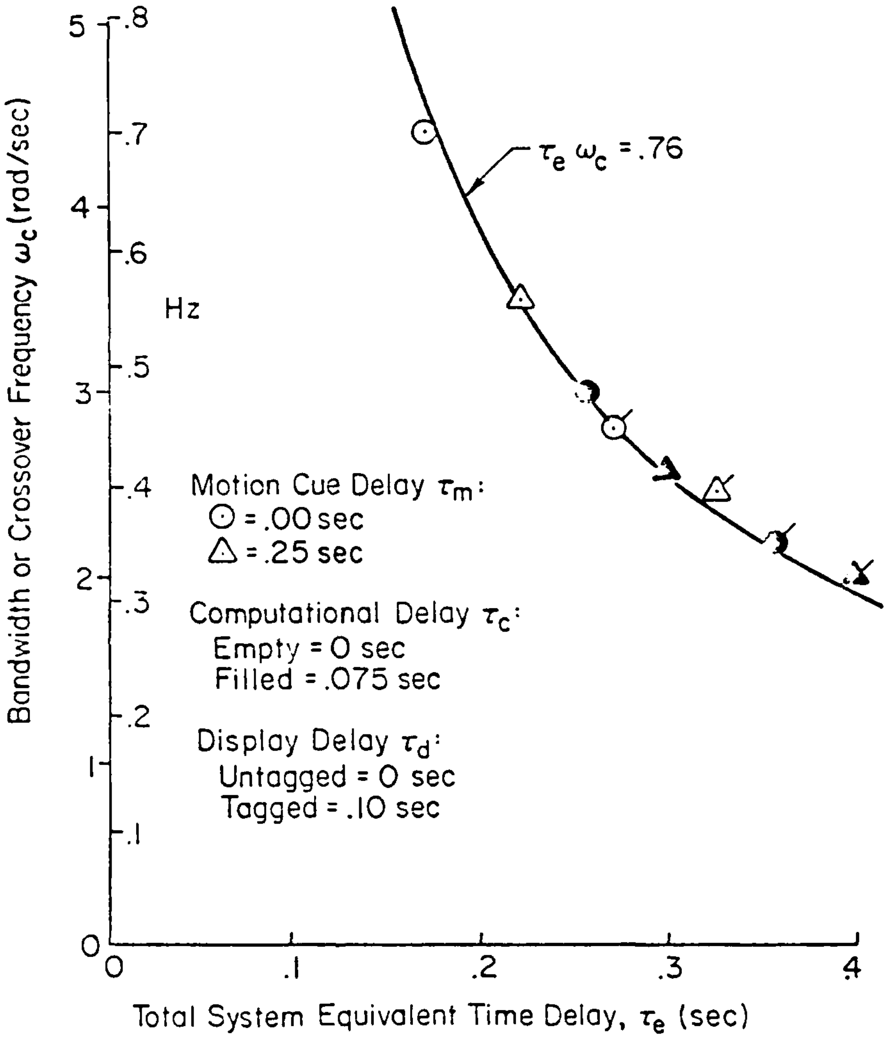


Figure B-7. System Bandwidth as a Function of System Time Delav

Thus, an increase of one third in the total effective system time delay (τ_e) would be acceptable. For exceptionally responsive real world systems, such as cars which can result in effective time delays on the order of 0.17 seconds (Ref. B-7), such an incremental increase in time delay due to simulator characteristics, would be on the order of 50 msec. (Maximum time delays on the order of 40 msec have previously been recommended for driving simulators, Ref. B-12.) For sluggish real world systems where effective system time delays might be 0.3-0.4 seconds, then incremental time delays on the order of 100 msec might be acceptable.

Regardless of the value of the constant in the Fig. B-7 relationship, the tradeoff between system bandwidth and effective system time delay is fundamental, and gives some insight into the consequences of added computational delays, whatever their origin.

H. PERFORMANCE EFFECTS

A δ_d impulse disturbance was applied to the Fig. B-3 model as indicated in order to investigate the performance consequences of various time delay sources. The impulse input might be attributable to a wind gust or road input in the case of ground vehicles. Time histories of the model transient response to an impulse disturbance input are illustrated in Fig. B-8 for an automobile traveling at U_o = 80 ft/sec (55 mph). For the low frequency kinematic characteristics given in Table B-2 (U_o/R = 0.2-0.92) the Fig. B-8 transients could also be scaled to represent airplane motions in the Fig. B-1 model (e.g., at 800 ft/sec this would represent target ranges of roughly 900-4000 ft).

The effects of the various transport delays on system performance are quite evident in Fig. B-8. Note that the model's ability to maintain lane position deteriorates radically as the amount of simulator delay is increased. The effect of the various delay sources are directly observable in the steering wheel response of the model driver. As the delay sources are concatenated, the model driver takes longer and longer to initially respond to the input disturbance. This is consistent with the data given in Table B-2 which shows the total effective

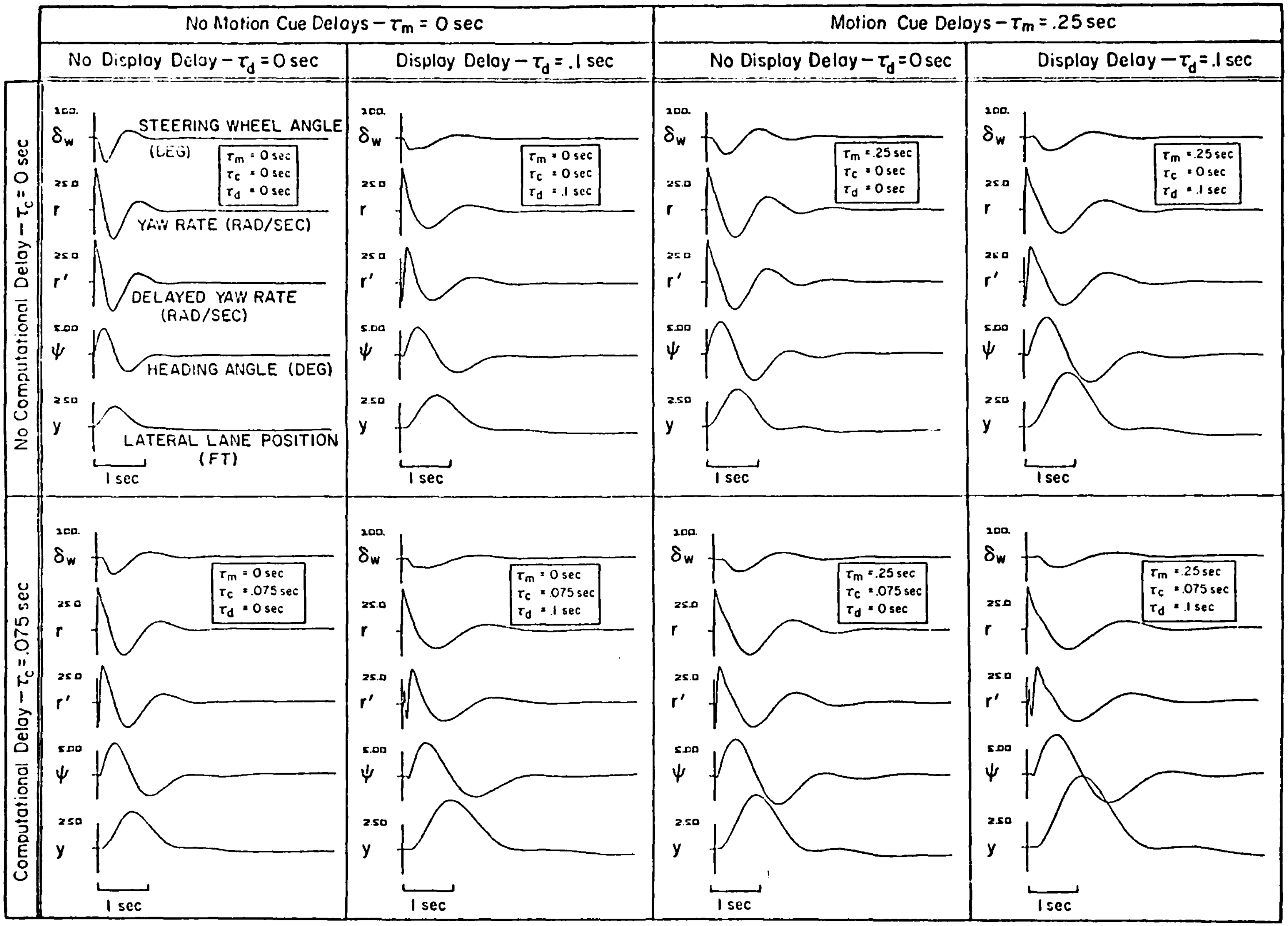


Figure B-8. Driver/Vehicle System Closed-Loop Response to an Impulse Disturbance system time delay increasing from a no delay level of 0.17 seconds to 0.38 seconds in the worst delay case.

The cycle time of the system transient response also obviously increases with increasing delay sources in Fig. B-8. This effect is consistent with the decreasing bandwidth as a function of time delay shown in Fig. B-7. Because of the driver/vehicle system's increasingly delayed regulatory response to the transient input, the maximum vehicle heading deviation nearly doubles in the worst delay case compared to the no delay condition, and the lane deviation increases by more than a factor of three with the increasing delay. Note also that each of the delay components considered separately in Fig. B-8 have a similar effect on system performance, as does the concatenation of any two delay sources.

I. SYSTEM ARCHITECTURE AND DELAY COMPENSATION

The effective system delays analyzed herein can arise from a variety of sources. Effective computational delays are due to a composite of A/D and D/A operations, computational algorithms (e.g., integration routines) and general software architecture. Cycle time may not be a true measure of effective delay if some routines are updated more often than others (e.g., high frequency modes might be updated more rapidly than kinematic integrations). Motion drive computations can have analogous considerations, and the frequency response of the drive servos must also be accounted for. CGI systems must maintain high refresh and update rates to portray smooth motion (i.e., typically 50 Hz or above), but multiple frame times may be required for angular and translational commands work their way through typical pipeline architectures.

Delay compensation can be considered at various stages in the system architecture. Minimum delay integration routines should be considered for dynamic computations (Ref. B-13). The update of motion and angular orientation cues are more critical to closed-loop operator/vehicle system response than outer loop translational information that is already delayed by kinematic integration. Thus in computing equations of motion, angular rates and orientation, and accelerations could be updated more rapidly than inertial velocity and position. In CGI display systems, angular transformations could be updated more rapidly than perspective transformations.

Lead or rate compensation might be considered for both host computer and CGI computations. Overall system dynamics should be considered here, however. The transfer functions in Figs. B-4 and B-6 suggest that for systems with adequate motion cues, lead frequencies in the region of the human operators limb/manipulator bandwidth (> 10 rad/sec) might be acceptable, while in the case of delayed or no motion cues, lead compensation could be increased to cover the bandwidth above the basic vehicle dynamics bandwidth. In general lead frequency must be above system crossover frequency (ω_c) in order to avoid compromising system gain margin.

J. CONCLUDING REMARKS

The model analysis herein shows that the effects of several computational delay sources in manual vehicle control systems can be evaluated to a first approximation by their effect on a composite effective system time delay. This effective time delay constrains the closed-loop bandwidth that can be achieved by the human operators. Tolerable computational delays can be determined by specifying a permissable system bandwidth reduction. The model analysis also shows that degradation in performance, such as regulation against transient disturbance, is consistent with system bandwidth reduction.

In general, compensation for effective system delays must be considered in an overall system context. System delays and compensation effects should be measured with input/output identification procedures using appropriate system inputs and sensors to measure outputs (e.g., gyros and accelerometers to measure platform motions and photo detectors to measure display system response). Response functions should be compensated to approach the less delayed response of the ideal target system. Finally, the fidelity of the system response should be considered from the human operator's point of view. In moving base systems, visual and motion cues should be consistent, and in general perceived vehicle response should be consistent with the operator's expectations. The analytic consequences of these fidelity considerations are not well understood, and typically would require final empirical tuneup.

APPENDIX B

REFERENCES

B-1. Puig, Joseph A., William T. Harris, and Gilbert L. Ricard, Motion in Flight Simulation: An Annotated Bibliography, NAVTRAEQUIPCEN IH-298, July 1978.

B-2. Semple, Clarence A., Robert T. Hennessy, Mark S. Sanders, et al., Aircrew Training Device Fidelity Features, AFHRL-TR-80-36, Feb. 1980.

B-3. Baron, Sheldon, Ramal Muralidharan, and David Kleinman, "Closed Loop Models for Analyzing the Effects of Simulator Characteristics," AIAA Paper 78-1592, presented at the AIAA Flight Simulation Technologies Conference, Arlington, TX, 18-20 Sept. 1978, pp. 138-148.

B-4. Parrish, Russell V., Burnell T. McKissick, and Billy R. Ashworth, "Comparison of Simulator Fidelity Model Predictions with In-Simulator Evaluation Data," NASA TP-2106, Feb. 1983.

B-5. Hess, Ronald A., "The Effects of Time Delays on Systems Subject to Manual Control," AIAA Paper 82-1523, presented at the AIAA Guidance and Control Conference Proceedings, San Diego, CA, 9-11 Aug. 1982, pp. 165-172.

B-6. Hoh, Roger H., Thomas T. Myers, Irving L. Ashkenas, et al., Development of Handling Quality Criteria for Aircraft with Independent Control of Six Degrees of Freedom, AFWAL-TR-81-3027, Apr. 1981.

B-7. Allen, R. Wade, "Stability and Performance Analysis of Automobile Driver Steering Control," SAE Paper 820303, presented at the 1982 SAE International Annual Congress and Exposition, Detroit, MI, 22-26 Feb. 1982.

B-8. Hodgkinson, J., W. J. LaManna, and J. L. Heyde, "Handling Qualities of Aircraft with Stability and Control Augmentation Systems -- a Fundamental Approach," Aeronautical Journal, Vol. 80, No. 782, Feb. 1976, pp. 75-81.

B-9. McRuer, Duane T., and Richard H. Klein, "Comparison of Human Driver Dynamics in an Automobile on the Road with Those in Simulators Having Complex and Simple Visual Displays," Systems Technology, Inc., P-173A, presented at the 55th Annual Meeting of the Transportation Research Board, Washington, D.C., 19-23 Jan. 1976.

B-10. Gum, D. R., and W. B. Albery, "Time-Delay Problems Encountered in Integrating the Advanced Simulator for Undergraduate Pilot Training," J. Aircraft, Vol. 14, No. 4, Apr. 1977, pp. 327-332.

B-11. Allen, R. Wade, "Modeling Driver Steering Control Behavior," Systems Technology, Inc., P-322, presented at the 1982 IEEE International Conference on Cybernetics and Society, Seattle, WA, 28-30 Oct. 1982.

B-12. Allen, R. Wade, and Henry R. Jex, "Driving Simulation -- Requirements, Mechanization and Application," SAE Paper 800448, presented at the SAE Congress and Exposition, Detroit, MI, Feb. 1980; also SAE Trans., Vol. 89, 1981, pp. 1769-1780.

B-13. Howe, R. M., "Special Considerations in Real-Time Digital Simulation," The Proceedings of the 1983 Summer Computer Simulation Conference, Volume 1, 11-13 July 1983, Vancouver, B.C., Canada.

APPENDIX C

A FAST, PROGRAMMABLE, LOW-COST DISPLAY DEVICE FOR MAN-IN-THE-LOOP SIMULATION

A. OVERVIEW

Throughput delays in man-in-the-loop simulation can cause degraded performance and lead to potentially unstable operation. Current CGI approaches generally have significant computational delays, and in addition involve a costly combination of a general purpose host computer and special purpose digital display processors. This appendix describes a microprocessor based calligraphic display system which includes an analog processor for accomplishing rotational and perspective transformations. The system permits defining a complete data base which can be rapidly mapped into instrument formats and a perspective plane for presenting out-the-window scenes for low-cost, vehicle control simulations.

B. *INTRODUCTION*

Visual display throughput delays can seriously degrade man-in-the-loop simulation performance (Refs. C-1 and C-2). In controlling responsive vehicles such as automobiles and high performance aircraft, the human operator represents a transport delay on the order 250 msec or less. Simulator feedback devices such as display systems should not significantly increase the total amount of closed loop transport delay in order to permit adequate performance and avoid stability problems (Ref. C-3).

The display approach described herein was developed as part of a low-cost simulator technology program, to provide a relatively high update rate calligraphic scene generator. It was designed around a microprocessor-based waveform generator in order to permit storage of complex, programmable three-dimensional data bases. An analog processor was designed to accomplish angular and perspective transformations in order to minimize cost and throughput delays. The basic digital wave

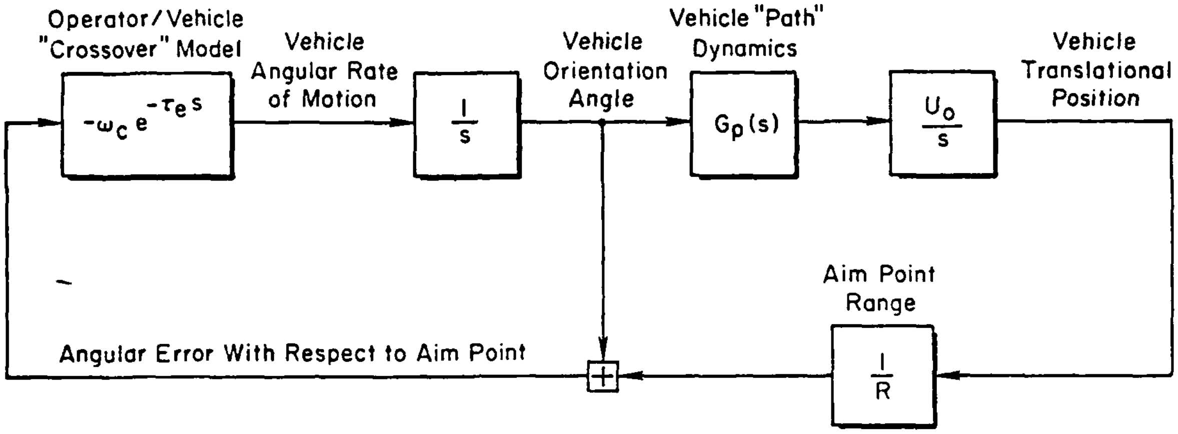
ω_c = Operator/Vehicle Bandwidth

τ_e = Effective Time Delay (operator + vehicle + computation)

$G_p(s)$ = Vehicle Path Dynamics + Computational Delay

U_o = Vehicle Forward Velocity

s = Laplace Transform Variable



Operator/Vehicle "Crossover" Model
$-\omega_c e^{-\tau_e s}$
Vehicle Angular Rate of Motion
$\frac{1}{s}$
Vehicle Orientation Angle
Vehicle "Path" Dynamics
$G_p(s)$
$\frac{U_o}{s}$
Vehicle Translational Position
Aim Point Range
$\frac{1}{R}$
–
Angular Error With Respect to Aim Point

$$\text{Phase Margin Stability Criteria}: \quad \phi_M = \frac{\pi}{2} - \tau_e \omega_c - \measuredangle\left[1 + \frac{U_o}{R} \cdot \frac{G_p(s)}{s}\right]$$

Figure C-1. Generic Model of a Human Operator/Vehicle Control Task to Illustrate the Effect of Computational Delays

form generator approach has previously been described (Ref. C-4). This appendix describes the throughput delay problem, and the architecture of a display system which provides a low-cost solution.

C. COMPUTATIONAL DELAY PROBLEM

A block diagram for a generic human operator/vehicle control task is illustrated in Fig. C-1. This simplified Laplace Transform dynamic model could represent a driver/car scenario (Ref. C-5) or a pilot/airplane control task (Ref. C-6). A "crossover" model form is used to approximate the combined response of the human operator and vehicle angular motion (Ref. C-7), which includes a gain, ω_c, and an equivalent time delay, τ_e. Aside from kinematic integrations, some extra dynamics are provided for the vehicle's path response. These dynamics are usually negligable for cars, and for airplanes amount to an equivalent first-order lag.

A simple control system stability criterion, termed phase margin, can be derived as indicated in Fig. C-1 (Ref. C-5). The system is stable for positive values of ϕ_M, but becomes more oscillatory the smaller ϕ_M gets. Note that ϕ_M decreases in direct proportion to the equivalent time delay, τ_e, of the driver/vehicle crossover characteristic. Thus, in a simulator, extra computational delays in vehicle dynamics and display systems directly affect stability.

It can be shown that delays in computing and/or displaying vehicle position do not affect stability as much as the τ_e delays. Extra delays in τ_e affect the feedback of vehicle attitude which the human operator senses almost immediately after control inputs. The vehicle position loop includes an additional integration, however, which makes feedback in this loop slow. Therefore time delays have relatively less influence on position changes than on attitude changes (Ref. C-8). This result has implications for display processor design, and suggests that delays in the angular orientation transformations should be minimized. The architecture for the processor described herein was designed from this point of view.

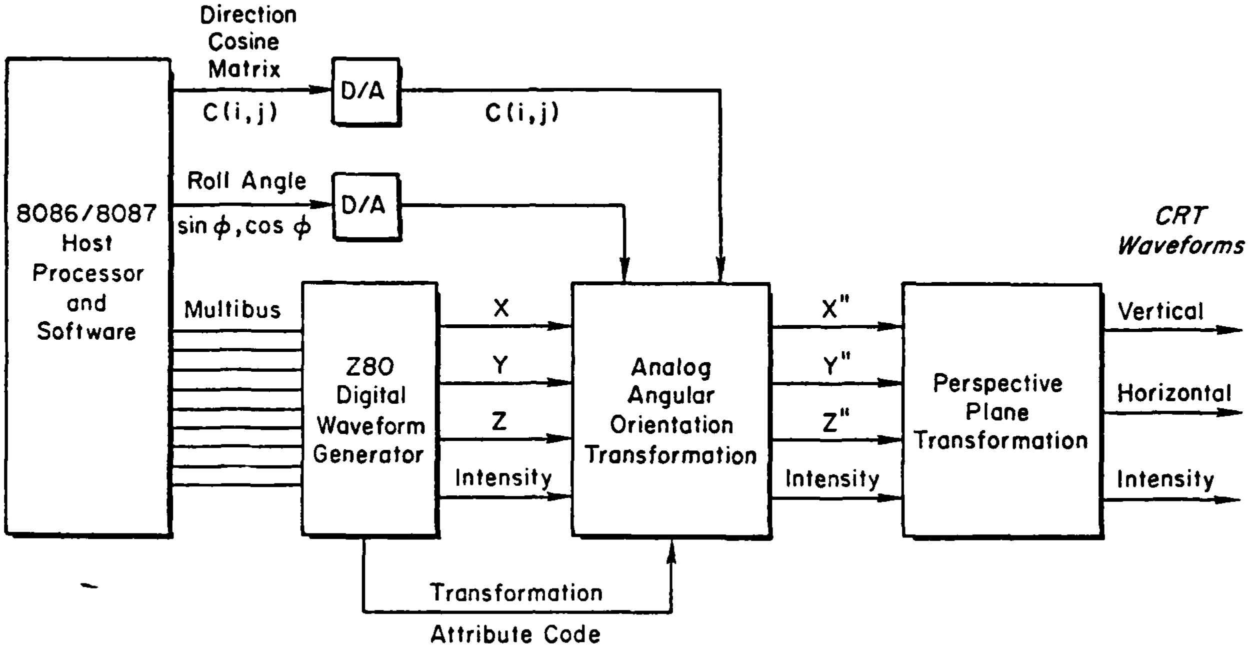
D. DISPLAY PROCESSOR DESIGN

The display system consists of two major elements: 1) a digital waveform generator which draws three-dimensional (rectangular coordinate) maps relative to observer position from memory stored coordinates; 2) an analog transformation system which provides angular transformations for orienting the observer's line of sight and a perspective transformation for drawing images within the observer's field of view on the display plane. The system architecture and software control is set up to accommodate instrument display formats which may require roll and/or pitch or no transformation, horizon scenes which require angular transformation but no perspective, and out-the-window scenes requiring all transformations.

A block diagram of the overall processor system is shown in Fig. C-2. The digital waveform generator consists of a single multibus card which contains a Z80 microprocessor, 64K bytes of memory (an arbitrary mix of RAM and ROM), and hybrid circuitry. This card generates x, y, and z axis analog waveforms, associated blanking pulses and an intensity control waveform, plus an attribute code which selects the transformations that will be applied to a given set of imagery. An Intel 8086 single board computer serves as the system host processor, and communicates with the Z80 card over the multibus through a shared memory scheme.

The 8086 can write directly into the shared Z80 memory space to cause different sets of imagery to be displayed and to update the observer's viewing position (x, y, z coordinates) within the three-dimensional map. Memory conflicts are arbitrated by a hardware bus lockout when the Z80 is reading, and by a shared software semaphore flag which the 8086 uses when it writes address data into Z80 RAM. The computational load on the Z80 is minimal, consisting of memory fetch and a fixed point add to update the observer position for each vector and point. Thus the vector rate of the processor is relatively high, running about 8000 vectors per second. At a 50 Hz update rate to minimize flicker, display scenes with about 160 vectors can be nominally presented. (For future upgrades, a faster 16 bit processor would allow more vectors and finer resolution.)

The digital waveform generator sends x, y, z, and intensity signals in the form of a three-dimensional vector to an analog transformation processor, which consists of angular and perspective transformation circuitry as indicated in Fig. C-2. The angular transformations are performed by a direction cosine matrix. The direction cosine matrix is computed by six degree-of-freedom vehicle equations of motion in the 8086 host processor using quaternions. The host processor card includes an 8087 hardware math coprocessor for maximum computational speed. The direction cosine matrix is then D/A converted and sent to the analog angular transformation processor as analog voltages. This approach means that angular motions exhibit minimal display processor delay.



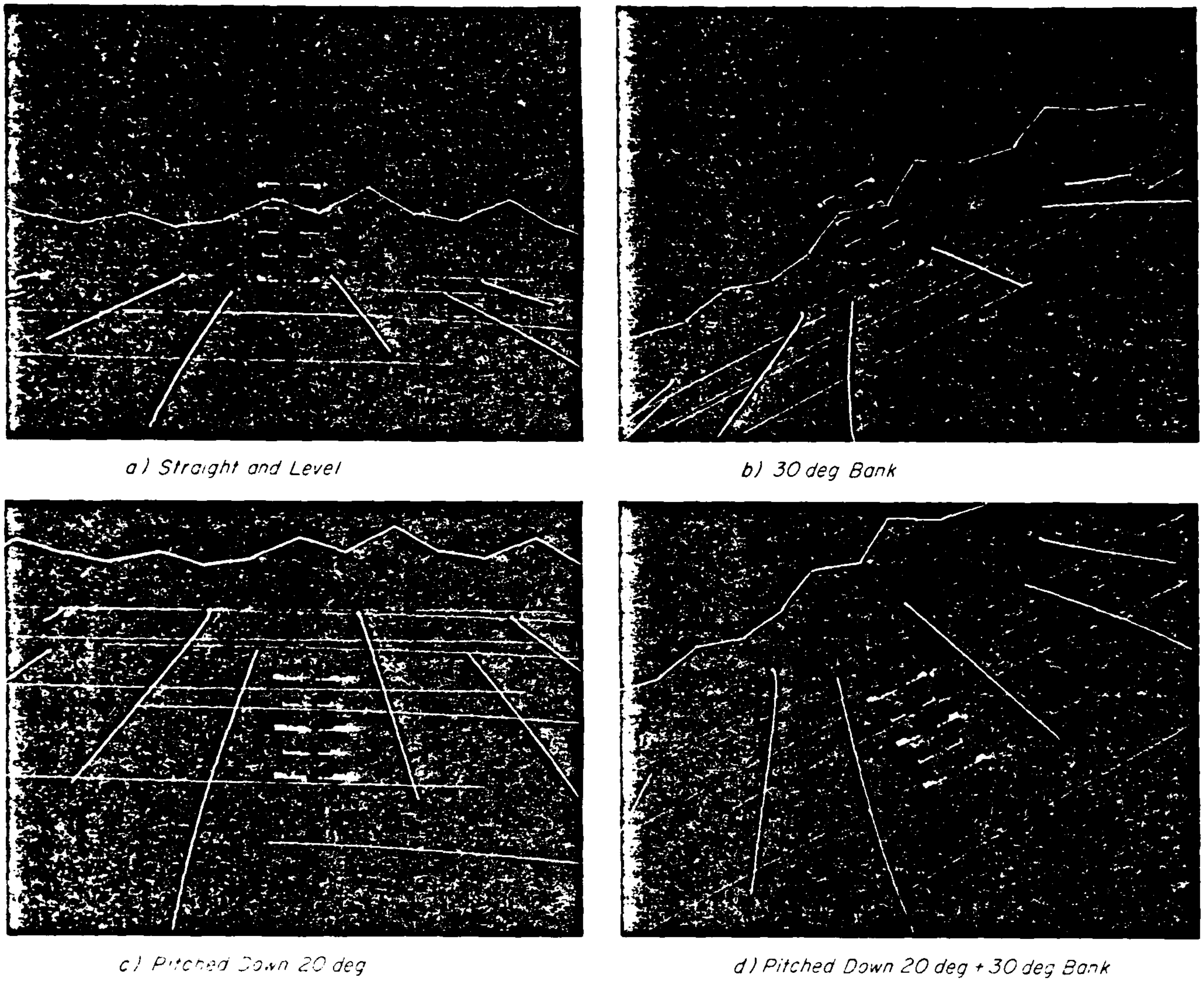
Direction Cosine Matrix
C(i,j)
D/A
C(i,j)
Roll Angle
sin ϕ, cos ϕ
D/A
8086/8087 Host Processor and Software
Multibus
Z80 Digital Waveform Generator
X
Y
Z
Intensity
Analog Angular Orientation Transformation
X"
Y"
Z"
Intensity
Perspective Plane Transformation
CRT Waveforms
Vertical
Horizontal
Intensity
Transformation Attribute Code

Figure C-2. Computer/Processor Architecture for Calligraphic Display System

E. CONCLUDING REMARKS

Small angular motion delays are an important feature of the display system described here for application to vehicle control simulations. Since the human operator's highest control bandwidth requirements are in attitude control, this control would be most degraded by computational delay. Another powerful feature of this approach to display generation is that each vector is separately encoded for the types of transformations to be applied to it by the transformation hardware. As noted in the Fig. C-2 block diagram, the Z80 microprocessor sends a transformation attribute code with each vector. The attribute code is then used by the transformation processor to determine which of several sets of transformations will be applied to the vectors. For example, this approach allows generation of HUD (Head-Up Display) pitch scales which only require roll transformation, airplane referenced vectors (e.g., gun tracers) which only go through the perspective transformation, and perhaps other HUD symbology that doesn't require any transformation at all.

Photos of several display configurations are shown in Fig. C-3. This graphics approach allows for relatively complex programmable scene generation with low computational delays. The hardware is inherently low-cost, and allows for additional future expansion of capabilities. This type of calligraphic system could potentially provide for curved vectors (not just straight line approximations) through additional analog transformation (see Ref. C-9) and could also include some limited fill capability by high frequency modulation of the vectors.



a) Straight and Level

b) 30 deg Bank

c) Pitched Down 20 deg

d) Pitched Down 20 deg + 30 deg Bank

Figure C-3. Calligraphic Display Scenes Including Horizon, Ground Grid, and HUD Pitch Scale

APPENDIX C

REFERENCES

C-1. Puig, Joseph A., William T. Harris, and Gilbert L. Ricard, Motion in Flight Simulation: An Annotated Bibliography, NAVTRAEQUIPCEN IH-298, July 1978.

C-2. Gum, D. R., and W. B. Albery, "Time-Delay Problems Encountered in Integrating the Advanced Simulator for Undergraduate Pilot Training," J. Aircraft, Vol. 14, No. 4, Apr. 1977, pp. 327-332.

C-3. Allen, R. Wade, and Henry R. Jex, "Driving Simulation -- Requirements, Mechanization and Application," SAE Paper 800448, presented at the SAE Congress and Exposition, Detroit, MI, Feb. 1980; also SAE Trans., Vol. 89, 1981, pp. 1769-1780.

C-4. Garret, Billy, "Microvec: The Other Type of Video Display," BYTE, Vol. 7, No. 11, Nov. 1982, pp. 508-528.

C-5. Allen, R. Wade, "Stability and Performance Analysis of Automobile Driver Steering Control," SAE Paper 820303, presented at the International Congress and Exposition, Detroit, MI, 22-26 Feb. 1982.

C-6. Hoh, Roger H., Thomas T. Myers, Irving L. Ashkenas, et al., Development of Handling Quality Criteria for Aircraft with Independent Control of Six Degrees of Freedom, AFWAL-TR-81-3027, Apr. 1981.

C-7. McRuer, D. T., and E. S. Krendel, Mathematical Models of Human Pilot Behavior, AGARDograph No. 188, Jan. 1974.

C-8. Allen, R. Wade, "Computational Considerations in Real Time Simulation Computer Graphics," Systems Technology, Inc., P-341, presented at the Computer Graphics '84, Conference and Exposition, Anaheim, CA, 13-17 May 1984.

C-9. Allen, R. Wade, Jeffrey R. Hogge, and Stephen H. Schwartz, "A Simulator for Research in Driver, Vehicle and Environment Interaction," Systems Technology, Inc., P-184, presented at the 56th Annual Meeting of the TRB, Washington, D.C., Jan. 1977.